Desain dan Implementasi *Peer-to-Peer Accommodation Platform* dengan Integrasi *Smart Contract*, *Self-Sovereign Identity*, dan *Zero Knowledge Proof*

Laporan Tugas Akhir

Disusun sebagai syarat kelulusan tingkat sarjana

Oleh

CEAVIN RUFUS DE PRAYER PURBA

NIM: 18221162

PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

JUNI 2025

Desain dan Implementasi *Peer-to-Peer Accommodation Platform* dengan Integrasi *Smart Contract*, *Self-Sovereign Identity*, dan *Zero Knowledge Proof*

Laporan Tugas Akhir

Oleh

CEAVIN RUFUS DE PRAYER PURBA

NIM: 18221162

Program Studi Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Telah disetujui dan disahkan sebagai Laporan Tugas Akhir

di Bandung, pada tanggal 17 Juli 2025

Pembimbing,

Ir. Budi Rahardjo, M.Sc., Ph.D.

NIP 109 110 001

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Pengerjaan dan penulisan Laporan Tugas Akhir ini dilakukan tanpa menggunakan bantuan yang tidak dibenarkan.
2. Segala bentuk kutipan dan acuan terhadap tulisan orang lain yang digunakan di dalam penyusunan laporan tugas akhir ini telah dituliskan dengan baik dan benar.
3. Laporan Tugas Akhir ini belum pernah diajukan pada program pendidikan di perguruan tinggi mana pun.

Jika terbukti melanggar hal-hal di atas, saya bersedia dikenakan sanksi sesuai dengan Peraturan Rektor ITB No. 257 tahun 2019 tentang Penegakan Norma Akademik dan Kemahasiswaan Institut Teknologi Bandung.

Bandung, 16 Juni 2025

Ceavin Rufus De Prayer Purba
NIM 18221162

ABSTRAK

Desain dan Implementasi *Peer-to-Peer Accommodation Platform* dengan Integrasi *Smart Contract*, *Self-Sovereign Identity*, dan *Zero Knowledge Proof*

Oleh

CEAVIN RUFUS DE PRAYER PURBA

NIM : 18221162

Industri pariwisata global terus mengalami pertumbuhan signifikan, diikuti dengan meningkatnya kebutuhan akan sistem pemesanan akomodasi daring yang aman dan menjaga privasi. Namun, model transaksi konvensional yang mengandalkan pihak ketiga untuk memverifikasi dan mengelola data pengguna tidak hanya rentan terhadap pelanggaran privasi, tetapi juga menciptakan ketergantungan yang tidak efisien. Permasalahan ini berakar pada arsitektur sistem yang tersentralisasi, yang mengharuskan pengguna menyerahkan informasi pribadi secara berlebihan untuk memperoleh hak akses. Menjawab tantangan tersebut, penelitian ini mengusulkan pendekatan alternatif melalui pengembangan sistem yang sepenuhnya anonim dan terdesentralisasi, dengan menjadikan identitas pengguna sebagai pusat kontrol. Anonimitas dalam konteks ini berarti bahwa data identitas pengguna tidak disimpan di dalam sistem sama sekali.

Menggunakan pendekatan *Design Science Research Methodology* (DSRM), penelitian ini merancang sistem *peer-to-peer accommodation platform* yang aman, anonim, dan terdesentralisasi. Sistem ini mengintegrasikan *smart contract* untuk otomatisasi transaksi, serta teknologi *self-sovereign identity* (SSI) dan *zero-knowledge proof* (ZKProof) untuk perlindungan privasi pengguna. Seluruh proses transaksi, termasuk pemesanan, pembatalan, dan penyelesaian sengketa, dijalankan melalui *smart contract* tanpa keterlibatan pihak ketiga, sehingga meningkatkan transparansi dan efisiensi. Implementasi SSI dan ZKProof memungkinkan pengguna tidak perlu mengungkapkan data pribadi, tetapi tetap dapat membuktikan hak akses melalui *booking credential* yang dikeluarkan sebagai bukti reservasi anonim. Untuk memastikan keunikan, kelayakan, dan keaktifan pengguna secara anonim, sistem menerapkan verifikasi terdesentralisasi berbasis bukti kriptografi.

Sebagai *proof of concept*, penelitian ini mengimplementasikan sistem yang dirancang untuk menunjukkan bahwa pendekatan yang dirancang berhasil

mewujudkan pemesanan akomodasi yang transparan, aman, dan sepenuhnya anonim. Evaluasi melalui pengujian fungsional, unit, dan *static analysis* pada *smart contract* mengonfirmasi bahwa seluruh fitur utama berjalan sesuai ekspektasi, dengan konsumsi *gas* yang efisien. Implementasi ini tidak hanya membuktikan kelayakan teknis dari arsitektur yang diusulkan, tetapi juga membuka peluang baru untuk pengembangan aplikasi identitas yang lebih menghormati privasi pengguna.

Kata kunci: *design science research*, *peer-to-peer*, *blockchain*, *smart contract*, *zero-knowledge proof*, *verifiable credential*, *self-sovereign identity*, *privacy-preserving identity*.

KATA PENGANTAR

Segala puji dan syukur saya panjatkan atas berkat dan kekuatan yang telah memampukan saya menyelesaikan tugas akhir yang berjudul "Desain dan Implementasi *Peer-to-Peer Accommodation Platform* dengan Integrasi *Smart Contract*, *Self-Sovereign Identity*, dan *Zero Knowledge Proof*". Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Sistem dan Teknologi Informasi, Institut Teknologi Bandung.

Tugas akhir ini tidak akan terselesaikan tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, saya ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Budi Rahardjo, M.Sc., Ph.D., selaku dosen pembimbing tugas akhir, atas bimbingan, arahan, dan waktu yang diberikan dengan penuh kesabaran selama proses penyusunan tugas akhir ini. Setiap saran dan koreksi beliau bagaikan peta dalam rimba berpikir yang rumit, membantu saya menemukan arah ketika kehilangan fokus.
2. Yudi Xu, yang telah menjadi mentor sekaligus rekan diskusi selama proses pengerjaan tugas akhir ini. Dukungan teknis, pemikiran kritis, dan pandangan praktis yang ia berikan telah melengkapi bimbingan akademik yang saya terima, serta menjadi jembatan penting antara teori dan implementasi di lapangan.
3. Seluruh staf pengajar dan *civitas academica* program studi Sistem dan Teknologi Informasi ITB, atas ilmu dan pengalaman yang sangat berharga selama masa studi saya. Terima kasih telah membuka cakrawala berpikir saya, bukan hanya dalam logika dan sistem, tetapi juga dalam menghargai proses pembelajaran itu sendiri.
4. Keluarga saya yang tercinta, terutama Ayah dan Ibu, atas kasih sayang, doa, serta dukungan moral dan material yang terus menguatkan saya dalam menempuh pendidikan ini. Tanpa mereka, saya mungkin hanya akan

menjadi algoritma yang tak pernah selesai dikompilasi. Gagap makna, tanpa fondasi cinta dan harapan.

5. Teman-teman terdekat, yang tidak dapat penulis sebutkan satu per satu, terutama sekelompok remaja yang tergabung dalam grup *Terverifikasi Kacuk* dan *Sembilan Naga*, yang secara konsisten telah menjadi sumber tawa di tengah kesibukan, sekaligus pengingat bahwa perjuangan ini tidak harus dilalui dengan wajah serius setiap saat.

Saya berharap tugas akhir ini dapat memberikan kontribusi yang positif bagi pengembangan ilmu pengetahuan dan menjadi dasar bagi pengembangan penelitian atau implementasi lebih lanjut di masa yang akan datang.

Bandung, 16 Juni 2025

Ceavin Rufus De Prayer Purba

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR RUMUS

BAB I

PENDAHULUAN

I.1 Latar Belakang

Industri pariwisata global terus mengalami pertumbuhan yang signifikan, dan di Indonesia, sektor ini memegang peranan penting dalam perekonomian nasional. Berdasarkan data Badan Pusat Statistik pada Agustus 2024, tercatat 1,34 juta wisatawan mancanegara dan 75,88 juta wisatawan domestik. Pertumbuhan sektor pariwisata ini didukung oleh transformasi digital yang memungkinkan kemunculan berbagai platform untuk mendukung sektor ini. Salah satu inovasi yang semakin populer adalah platform *peer-to-peer accommodation platform* (P2P AP), yang memungkinkan wisatawan terhubung langsung dengan penyedia akomodasi.

Sebagian besar P2P AP saat ini menggunakan *centralized architecture*, yang mengelola data pribadi seperti identitas, informasi pembayaran, dan preferensi pengguna oleh satu entitas tunggal (Rosoon, Choksuchat, dan Aiyarak 2023; Raj 2019). Namun, pendekatan ini rentan terhadap berbagai risiko, seperti kebocoran data, manipulasi informasi, dan potensi penyalahgunaan oleh pihak yang tidak bertanggung jawab. Kasus penghapusan ulasan negatif oleh beberapa P2P AP (Rosoon, Choksuchat, dan Aiyarak 2023; Blengini dan Venturini 2024) menunjukkan potensi penyalahgunaan data yang dikelola secara terpusat. Keamanan aplikasi sangat mempengaruhi tingkat kepercayaan pengguna, terutama pada platform yang berkaitan dengan transaksi digital, yang semakin menuntut solusi yang lebih aman dan tepercaya (Rabbani dkk. 2023; Agag dan Eid 2019).

Untuk mengatasi tantangan tersebut, teknologi kriptografi menjadi dasar dari berbagai pendekatan solusi yang berfokus pada peningkatan keamanan dan privasi. Beberapa alternatif teknologi yang berbasis kriptografi meliputi integrasi *smart contract*, *zero-knowledge proof* (ZKProof), dan *self-sovereign identity* (SSI);

pemanfaatan *non-fungible token* (NFT) untuk bukti transaksi; serta penggunaan *homomorphic encryption* dan *secure multi-party computation* (SMPC) untuk menjaga kerahasiaan data selama proses transaksi. Setiap pendekatan ini menawarkan cara berbeda dalam memastikan bahwa informasi pengguna tetap terlindungi tanpa mengorbankan transparansi dan fungsionalitas platform.

Salah satu teknologi yang menonjol di antara pendekatan tersebut adalah *blockchain*, yang memungkinkan terbentuknya *trust-free system*. Pada sistem ini, kepercayaan antar *peers* tidak lagi menjadi kendala utama dalam melakukan transaksi (Dann dkk. 2020). Teknologi ini memiliki sifat *decentralized*, transparansi, dan tidak dapat diubah (*immutable*), sehingga menjadi alternatif yang lebih aman dibandingkan dengan sistem tradisional. Transparansi *blockchain* juga memungkinkan semua transaksi diverifikasi secara terbuka tanpa memerlukan perantara, sehingga meningkatkan kepercayaan pengguna terhadap platform (Nakamoto 2008).

Selain itu, konsep *self-sovereign identity* (SSI) muncul sebagai pendekatan baru yang memberikan individu kendali penuh atas identitas digital mereka, tanpa bergantung pada *database* terpusat (Raipurkar dkk. 2023). SSI memungkinkan pengguna untuk memverifikasi identitas atau *credential* mereka secara selektif, tanpa harus mengungkapkan seluruh informasi pribadi. Dengan demikian, pengguna dapat bersifat anonim, dalam artian data pengguna tidak perlu disimpan pada platform untuk melakukan transaksi atau interaksi digital, selama dapat membuktikan keabsahan klaim mereka secara kriptografis.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan teknologi *blockchain* untuk menciptakan *trust-free system* pada platform P2P AP. Dengan memanfaatkan keunggulan *blockchain*, diharapkan solusi ini dapat meningkatkan kepercayaan pengguna, melindungi privasi mereka, serta menghadirkan pengalaman yang lebih aman, efisien, dan terpercaya dalam industri pariwisata yang terus berkembang.

I.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, dirumuskan beberapa masalah utama yang diselesaikan dalam penelitian ini.

1. Bagaimana cara merancang sistem transaksi pada P2P AP yang meminimalkan keterlibatan pihak ketiga guna meningkatkan transparansi dan keamanan?
2. Bagaimana cara merancang sistem pemesanan anonim pada P2P AP tanpa menyimpan informasi pengguna?
3. Bagaimana membangun mekanisme verifikasi pengguna yang dapat mencegah penyalahgunaan dalam sistem pemesanan anonim tanpa mengorbankan privasi pengguna?

I.3 Tujuan

Berdasarkan masalah yang telah dirumuskan, dibuat tujuan yang ingin dicapai dalam pelaksanaan penelitian ini.

1. Merancang mekanisme transaksi pada P2P AP yang meminimalkan keterlibatan pihak ketiga guna meningkatkan transparansi dan keamanan.
2. Mengembangkan sistem pemesanan anonim pada P2P AP yang memungkinkan pengguna melakukan reservasi tanpa perlu menyimpan informasi identitas secara langsung.
3. Membangun sistem verifikasi pengguna yang mampu mencegah penyalahgunaan dalam sistem pemesanan anonim tanpa mengorbankan privasi pengguna.

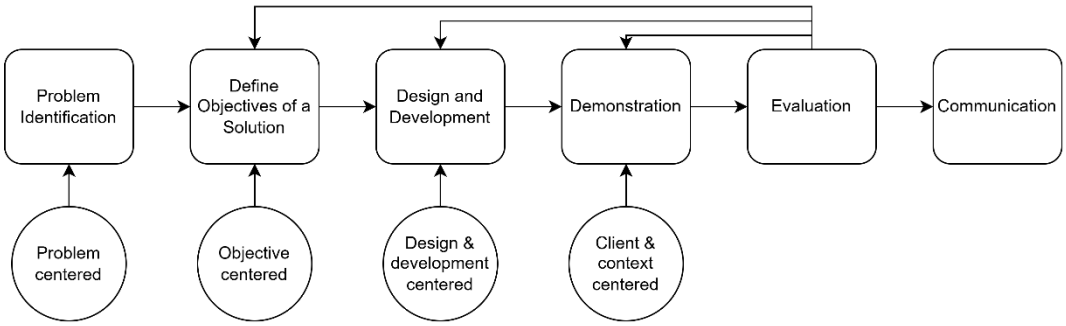
I.4 Batasan Masalah

Penelitian ini berfokus pada implementasi *smart contract*, *zero-knowledge proof*, dan *self-sovereign identity* dalam P2P AP. Pembahasan mengenai antarmuka pengguna tidak akan dilakukan secara mendalam, meskipun antarmuka tersebut

tetap dikembangkan sebagai sarana interaksi dan demonstrasi terhadap fungsionalitas sistem. Penelitian ini juga tidak membahas atau mengembangkan sistem *know your customer* (KYC). Sistem hanya akan memverifikasi bukti bahwa pengguna telah melakukan KYC melalui pihak ketiga yang terpercaya (*KYC provider*), serta melakukan verifikasi terhadap keunikan dan keaktifan pengguna menggunakan pendekatan *zero-knowledge proof*. Selain itu, penelitian ini juga akan menggunakan *identity wallet* dari pihak ketiga, sehingga pembahasannya tidak dilakukan secara mendalam.

I.5 Metodologi

Metodologi yang digunakan dalam penelitian penelitian ini adalah pendekatan *Design Science Research Methodology* (DSRM), yang pertama kali diperkenalkan oleh Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, dan S. Chatterjee pada tahun 2007 (Venable, Pries-Heje, dan Baskerville 2017) sebagai salah satu jenis metodologi Design Science Research (DSR). Metodologi DSRM adalah pendekatan yang digunakan untuk merancang dan mengembangkan solusi praktis terhadap masalah yang relevan, serta menguji dan mengevaluasi hasil desain tersebut. DSRM sangat sesuai untuk proyek yang bertujuan untuk menciptakan dan mengimplementasikan teknologi baru (Osvaldo dan Sordi 2021), dalam hal ini teknologi *blockchain* pada P2P AP. DSRM terdiri dari beberapa tahapan yang harus dilalui untuk menyelesaikan suatu masalah melalui desain, seperti yang ditunjukkan pada Gambar I.1.



Gambar I.1 Gambaran besar tahapan DSRM (Peffers dkk. 2007)

1. Identifikasi Masalah

   Penelitian ini dimulai dengan mengidentifikasi masalah yang ada dalam P2P AP tradisional. Pengumpulan fakta dilakukan dengan cara menggali informasi latar belakang, mengidentifikasi masalah utama yang ada dalam sistem tradisional, serta mengumpulkan data dari berbagai sumber, seperti buku, *conference paper*, berita, dan jurnal. Langkah ini bertujuan untuk memberikan gambaran yang jelas tentang konteks masalah yang akan diselesaikan.

2. Definisi Tujuan

   Perumusan tujuan dilakukan berdasarkan hasil identifikasi masalah dan potensi solusi yang dapat diterapkan secara praktis. Untuk memastikan relevansi dan keberhasilan implementasi, kriteria desain akan dirumuskan dengan mempertimbangkan indikator kuantitatif, seperti pengurangan biaya transaksi atau waktu pemrosesan, serta indikator kualitatif, seperti peningkatan kepercayaan pengguna. Setiap tujuan dirancang secara spesifik dan terukur, sehingga memudahkan proses evaluasi efektivitas solusi yang dihasilkan berdasarkan analisis masalah dan kajian terhadap solusi yang ada.

3. Pengembangan Solusi

   Artefak solusi dirancang dengan mengacu pada kebutuhan yang telah diidentifikasi. Tahapan ini meliputi desain fungsi utama dan struktur sistem, serta pengembangan implementasi yang sesuai dengan kriteria yang telah ditentukan. Artefak dapat berupa model kerja, metode, atau sistem implementasi yang menggambarkan solusi inovatif terhadap masalah. Proses iteratif diterapkan untuk memastikan penyempurnaan artefak, menggunakan prinsip-prinsip teori yang relevan sebagai acuan. Pendekatan ini juga memungkinkan penyelarasan artefak dengan kebutuhan spesifik dari konteks penelitian.

4. Demonstrasi

   Setelah solusi selesai dikembangkan, tahap demonstrasi dilakukan untuk menguji kemampuan sistem dalam hal fungsionalitas dan efektivitasnya. Simulasi atau uji coba akan dilakukan dengan menggunakan data yang relevan untuk memperlihatkan cara solusi yang dirancang mampu mengatasi permasalahan yang telah diidentifikasi. Hasil dari proses ini memberikan gambaran mengenai kinerja artefak yang dihasilkan dalam konteks aplikasi nyata.

5. Evaluasi

   Tahap evaluasi dilakukan untuk mengukur sejauh mana solusi yang dikembangkan berhasil mencapai tujuan yang telah ditentukan. Proses ini melibatkan pengumpulan data dari hasil pengujian pada tahap demonstrasi, disertai umpan balik dari pengguna atau pakar yang relevan. Metode evaluasi yang digunakan dapat mencakup analisis kinerja, survei pengguna, serta wawancara untuk memperoleh informasi mengenai efektivitas dan potensi kelemahan solusi yang dirancang.

6. Komunikasi

   Pada tahap akhir, hasil penelitian disampaikan kepada komunitas ilmiah dan praktisi melalui laporan yang komprehensif. Laporan ini merangkum seluruh proses penelitian, mulai dari identifikasi masalah, pengembangan solusi, evaluasi yang dilakukan, hingga hasil yang diperoleh. Selain itu, laporan tersebut juga mencakup rekomendasi untuk penerapan solusi dalam skala yang lebih luas atau usulan penelitian lanjutan yang dapat dikembangkan di masa depan.

BAB II

STUDI LITERATUR

Bab ini membahas landasan teori dan kajian literatur yang relevan untuk mendukung penelitian ini. Studi literatur dilakukan untuk memahami konsep, teori, dan teknologi yang mendasari penelitian, termasuk P2P AP, *blockchain*, *zero-knowledge proofs*, dan *self-sovereign identity*. Selain itu, kajian terhadap penelitian terdahulu yang berkaitan juga dilakukan untuk mengidentifikasi solusi yang sudah ada, celah penelitian, serta pendekatan yang dapat diadopsi atau dikembangkan lebih lanjut. Hasil dari studi literatur ini akan menjadi dasar dalam merumuskan solusi desain yang diusulkan dalam penelitian ini.

II.1 *Peer-to-Peer Accommdation Platform* (P2P AP)

P2P AP merupakan salah satu bentuk *sharing economy* yang berbasis *online platform*. Platform ini memungkinkan individu untuk menyewakan ruang yang tersedia dalam waktu singkat, baik berupa sebagian ruang di properti mereka maupun keseluruhan properti. Properti sendiri dapat berupa rumah ataupun hotel. (Česnuitytė dkk. 2022). Pada P2P AP, seorang pengguna dapat berperan menjadi *guest* (tamu) dan/atau menjadi *host*. *Guest* adalah orang yang menyewa ruangan atau properti dari *host* untuk jangka waktu tertentu, sedangkan *host* adalah pemilik properti yang menyewakan ruang atau properti tersebut kepada *guest*.

Platform P2P AP dapat dibedakan menjadi dua model bisnis utama, yaitu komersial yang berorientasi *profit* dan non-komersial yang lebih fokus pada aspek sosial. Airbnb adalah contoh populer dari P2P AP komersial, sedangkan Couchsurfing lebih condong ke arah non-komersial. Penelitian ini berfokus pada P2P AP dengan model bisnis komersial, yang melibatkan pertukaran nilai ekonomi secara eksplisit dalam transaksi antara penyedia dan pengguna layanan. Model seperti ini menekankan pada aspek monetisasi, efisiensi transaksi, serta mitigasi risiko. Oleh

karena itu, penelitian ini memberikan perhatian khusus terhadap mekanisme pembayaran, sistem reputasi, dan jaminan keamanan yang mendukung operasional platform komersial.

Meskipun fokus utama penelitian berada pada konteks komersial, prinsip-prinsip yang dikembangkan memiliki potensi untuk diterapkan dalam model non-komersial. Pada platform non-komersial seperti Couchsurfing, yang lebih menekankan pada pertukaran sosial dan interaksi budaya, kebutuhan akan keamanan, privasi, dan keaslian identitas pengguna tetap menjadi aspek yang penting. Dengan demikian, temuan dalam penelitian ini dapat memberikan kontribusi yang relevan bagi pengembangan ekosistem P2P AP non-komersial, dengan menyesuaikan pendekatan teknis dan tujuan penggunaannya.

II.2 *Digital Identity*

Digital identity merujuk pada representasi dari identitas seseorang atau entitas yang dikelola dan diproses dalam dunia digital. Konsep ini melibatkan informasi, atribut, dan karakteristik yang digunakan untuk membuktikan atau memverifikasi identitas di dunia maya. *Digital identity* bisa berupa informasi pribadi seperti nama, alamat email, nomor telepon, hingga biometrik seperti sidik jari atau pengenalan wajah. Identitas digital sangat penting dalam konteks keamanan siber, transaksi online, media sosial, *e-commerce*, dan banyak aplikasi lainnya. Wang dan De Filippi (2019) memaparkan perbedaan identitas, persona, atribut, dan *identifier*, yang seringkali seringkali saling tumpang tindih namun memiliki peran yang berbeda dalam sistem manajemen identitas digital (Wang dan De Filippi 2019).

Identitas mencakup semua atribut seseorang yang secara unik mendefinisikan individu tersebut sepanjang hidupnya, memberikan kesamaan dan kontinuitas meskipun aspek dan kondisi yang bervariasi. Identitas dapat mencakup sifat psikologis, budaya, sejarah, agama, dan tradisi yang menjadi bagian dari individu tersebut. Identitas bersifat dinamis dan multifaset, berkembang seiring waktu dan interaksi dengan lingkungan.

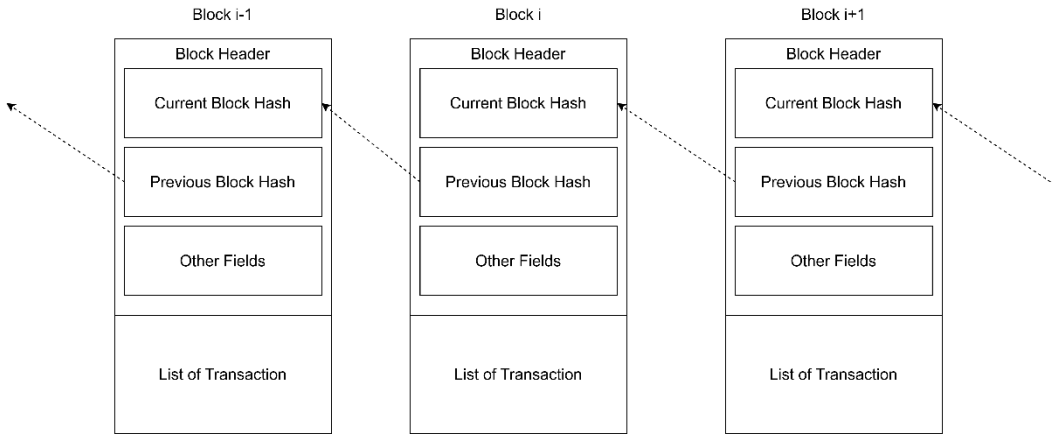
Persona adalah aspek spesifik dari identitas yang diekspresikan dalam konteks tertentu. Seseorang dapat memiliki beberapa persona tergantung pada konteks sosial yang dihadapi. Misalnya, seseorang bisa menjadi ibu yang berdedikasi di rumah, teman yang dapat dipercaya di antara teman-temannya, dan manajer yang tegas di tempat kerja. Persona adalah komponen penting dari sistem manajemen identitas karena berhubungan dengan cara individu mengautentikasi diri mereka ke dalam sistem.

Atribut menggambarkan properti esensial dan definisional dari seseorang yang memenuhi syarat sebagai anggota dari kelompok tertentu. Atribut tidak unik untuk individu tersebut dan dapat mencakup elemen seperti jenis kelamin, tinggi badan, kewarganegaraan, dan lain-lain. Atribut digunakan untuk mengklasifikasikan orang ke dalam kategori tertentu dan tidak dimaksudkan untuk mengidentifikasi individu secara unik.

Identifier adalah referensi yang digunakan untuk mengidentifikasi identitas dunia nyata atau persona tertentu. Identifier sering kali diberikan oleh pihak ketiga dan dapat berupa nama hukum, nomor jaminan sosial, atau nama pengguna. Identifier tidak dimaksudkan untuk menggambarkan atau memenuhi syarat seseorang, tetapi untuk mengidentifikasi seseorang dalam domain tertentu. Identifier harus unik dan tidak ambigu dalam domain tersebut.

II.3 *Blockchain*

Blockchain merupakan sebuah buku besar digital yang dikelola secara kolektif oleh jaringan komputer yang *decentralized*. Data pada *blockchain* disimpan dan didistribusikan di seluruh *node* jaringan, tidak bergantung pada *server* tunggal seperti dalam sistem terpusat. Setiap blok dalam rantai ini berisi kumpulan data transaksi yang telah diverifikasi dan dijamin keamanannya melalui kriptografi (Nakamoto 2008). Arsitektur *blockchain*, seperti yang ditunjukkan pada Gambar II.1, membuat *blockchain* sangat sulit untuk memanipulasi atau menghapus data setelah tercatat, sehingga integritas dan transparansi data terjaga.



Gambar II.1 Arsitektur *blockchain* (Nakamoto 2008)

Transaksi baru dicatat dalam blok yang dirantai secara kronologis, dengan validasi oleh seluruh *node* melalui mekanisme konsensus. Proses ini memastikan integritas dan transparansi data, serta menjadikan *blockchain* aman dari manipulasi, karena memerlukan kendali atas sebagian besar jaringan untuk melakukan perubahan, yang sangat sulit dilakukan (Nakamoto 2008).

II.3.1 Ethereum

Ethereum adalah platform terdesentralisasi yang memungkinkan pengembang untuk membangun *smart contract* dan aplikasi terdesentralisasi (dApps). Platform ini diusulkan oleh Vitalik Buterin pada tahun 2013. Sejak diluncurkan pada tahun 2015 sampai dokumen ini dibuat, Ethereum menjadi salah satu platform *blockchain* paling terkemuka. *Blockchain* pada Ethereum adalah buku besar publik yang mencatat semua transaksi dan eksekusi *smart contract* (Wood 2022).

Secara umum, ada tiga jenis aplikasi yang dapat dikembangkan di atas Ethereum (Buterin 2014):

1. aplikasi keuangan, yaitu aplikasi yang dirancang untuk membantu pengguna mengelola dan menandatangani kontrak menggunakan uang mereka. Contohnya adalah *sub-currencies*, wasiat, dan kontrak kerja;
2. aplikasi semi-keuangan, yaitu aplikasi yang melibatkan uang, tetapi juga memiliki sisi *non-monetary* yang signifikan. Contohnya adalah *self-*

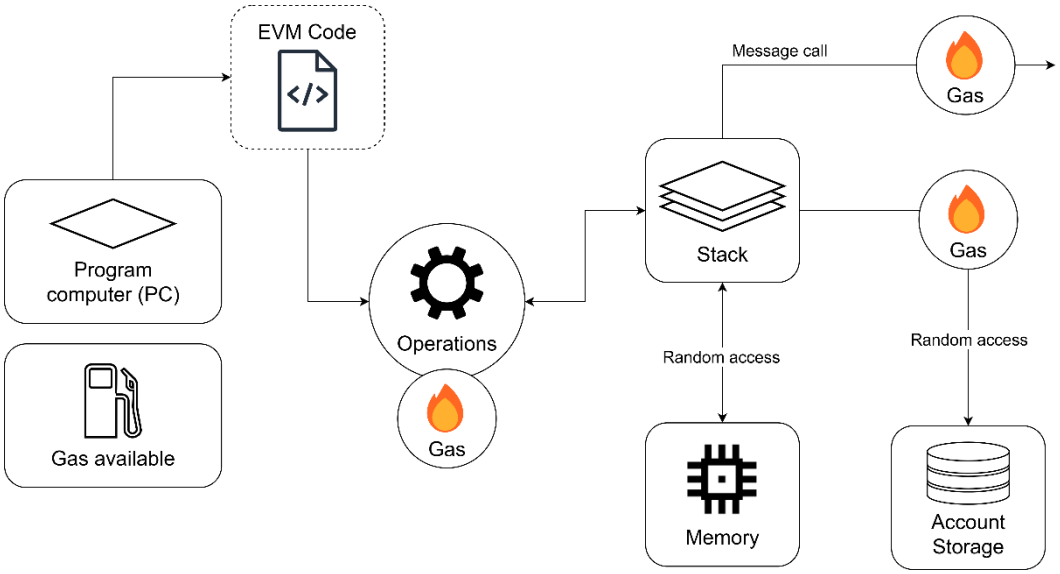
enforcing bounties yang memberikan insentif kepada individu untuk memecahkan masalah komputasi tertentu;

3. aplikasi non-keuangan, yaitu aplikasi yang tidak melibatkan aspek keuangan sama sekali. Contohnya adalah *online voting*.

II.3.2 *Ethereum Virtual Machine* (EVM)

EVM adalah lingkungan virtual terdesentralisasi yang digunakan untuk mengeksekusi *smart contract* di seluruh *node* Ethereum. Proses eksekusi ini dilakukan secara konsisten dan aman di seluruh jaringan, dan setiap operasi dalam *smart contract* membutuhkan *gas* untuk mengukur upaya komputasi yang dibutuhkan, sehingga memastikan alokasi sumber daya yang efisien dan keamanan jaringan (Ethereum.org 2023).

Alur eksekusi dalam EVM, yang terlihat pada Gambar II.2, dimulai dengan EVM *code* (*smart contract bytecode*) yang dieksekusi instruksi per instruksi berdasarkan posisi yang ditunjukkan oleh *program counter* (PC). Setiap instruksi atau operasi akan mengkonsumsi *gas*, yang memastikan penggunaan sumber daya komputasi berjalan efisien. Nilai-nilai sementara disimpan dalam *stack*, sedangkan data yang diperlukan selama eksekusi ditempatkan di *memory*. Jika diperlukan penyimpanan permanen, data disimpan di *account storage*, namun dengan biaya *gas* yang lebih tinggi. Selama eksekusi, *smart contract* dapat memanggil fungsi lain atau berinteraksi melalui *message calls*, yang mungkin memerlukan tambahan *gas*. Jika *gas* habis sebelum instruksi selesai, eksekusi akan gagal (Solidity 2024; Antonopoulos dan Wood Ph 2019; Ethereum.org 2023).



Gambar II.2 Alur eksekusi dalam EVM (Ethereum.org 2023)

Dengan arsitektur ini, EVM memastikan bahwa setiap operasi berjalan secara konsisten dan aman di semua node Ethereum, mendukung eksekusi *smart contract* yang terdesentralisasi dan transparan.

II.3.3 Solidity

Solidity adalah bahasa pemrograman tingkat tinggi yang dirancang khusus untuk pengembangan *smart contract* pada Ethereum. Solidity memungkinkan pengembang untuk menulis logika yang dieksekusi di EVM (Solidity 2024).

Solidity pertama kali diperkenalkan pada Agustus 2014 oleh Gavin Wood, salah satu pendiri Ethereum (Antonopoulos dan Wood Ph 2019). Bahasa ini terinspirasi dari C++, Python, dan JavaScript (Solidity 2024), sehingga memiliki sintaks yang mudah dipahami oleh pengembang dengan latar belakang pemrograman serupa. Solidity digunakan untuk menulis *smart contract* yang berisi aturan, logika bisnis, atau fungsi yang dijalankan secara otomatis di *blockchain* tanpa memerlukan perantara.

II.3.4 *Cryptocurrency*

Cryptocurrency adalah aset digital yang menggunakan teknologi kriptografi untuk mengamankan transaksi, mengontrol penciptaan unit baru, dan memverifikasi transfer aset tanpa perantara pihak ketiga seperti bank atau lembaga keuangan (Nakamoto 2008). Teknologi ini pertama kali diperkenalkan melalui Bitcoin oleh Satoshi Nakamoto pada tahun 2008, yang kemudian menjadi dasar bagi perkembangan ribuan jenis *cryptocurrency* lainnya.

Cryptocurrency beroperasi pada teknologi *blockchain*. Setiap transaksi diverifikasi oleh jaringan node melalui mekanisme konsensus, seperti *Proof-of-Work* (Nakamoto 2008) atau *Proof-of-Stake* 2(Kanani, Nailwal, dan Arjun 2019), yang memastikan integritas data tanpa memerlukan otoritas pusat. Teknologi ini memungkinkan transaksi dilakukan secara *peer-to-peer* (P2P), mengurangi biaya dan waktu yang biasanya dibutuhkan dalam sistem keuangan tradisional (Raj 2019).

Salah satu keunggulan utama *cryptocurrency* adalah sifatnya yang terdesentralisasi (Nakamoto 2008). Tidak ada entitas tunggal yang mengontrol jaringan, sehingga risiko kegagalan sistem atau penyalahgunaan kekuasaan dapat diminimalkan. Dengan fitur-fitur ini, *cryptocurrency* menawarkan solusi inovatif untuk sistem pembayaran, penyimpanan aset, dan transfer nilai yang lebih efisien serta inklusif.

II.3.5 *Smart Contract*

Smart contract adalah sebuah protokol yang dirancang untuk secara otomatis menjalankan, menegosiasikan, atau menegakkan perjanjian ketika kondisi tertentu terpenuhi (Raj 2019). Teknologi *blockchain* memungkinkan *smart contract* beroperasi secara terdesentralisasi tanpa melibatkan pihak ketiga, serta menjadikannya tahan terhadap perubahan atau manipulasi. Hal ini membuat *smart contract* menjadi solusi yang efisien untuk mengotomatisasi dan mengamankan berbagai transaksi (Bhushan dkk. 2021).

Keunggulan utama *smart contract* adalah tidak adanya potensi kesalahan manusia atau interpretasi yang keliru karena kontrak ini berjalan sesuai dengan kode yang

telah ditentukan. Selain itu, *smart contract* menawarkan transparansi karena dapat diaudit oleh semua pihak terkait (Raj 2019). Dengan proses yang sepenuhnya otomatis, *smart contract* juga membantu menghemat waktu dan sumber daya.

II.4 *Trust*

Trust merupakan elemen fundamental dalam interaksi antarpengguna pada sistem P2P, terutama ketika pengguna saling bertransaksi tanpa mengenal satu sama lain secara langsung. Menurut Mayer, Davis, dan David Schoorman (1995), *trust* didefinisikan sebagai kemauan satu pihak untuk menjadi rentan terhadap tindakan pihak lain berdasarkan harapan bahwa pihak tersebut akan bertindak sesuai dengan kepentingan si pemberi *trust*, tanpa adanya kontrol langsung. Dalam konteks digital, definisi ini diperluas mencakup kepercayaan terhadap sistem, algoritma, serta mekanisme pengambilan keputusan berbasis teknologi.

Dalam platform P2P, *trust* menjadi elemen krusial karena pengguna berinteraksi langsung tanpa keterlibatan institusi tradisional sebagai perantara. Dalam konteks sistem *social-technical*, *trust* memiliki dimensi yang kompleks dan multidimensional. Xu dkk. (2014) mengidentifikasi tiga jenis *trust* utama yang penting untuk mencapai hasil sistem yang optimal:

- kepercayaan interpersonal (antara pengguna),
- kepercayaan institusional (terhadap platform), dan
- kepercayaan terhadap teknologi (terhadap sistem digital yang digunakan).

Kepercayaan interpersonal berkaitan dengan sejauh mana pengguna mempercayai lawan transaksinya, sedangkan kepercayaan institusional dan teknologi berkaitan dengan persepsi terhadap keandalan platform, yang mencakup keamanan data, stabilitas sistem, dan efektivitas kebijakan penyelesaian sengketa. Keberadaan *trust* dalam ketiga dimensi ini memungkinkan pengguna merasa aman dan yakin untuk melakukan transaksi, berbagi sumber daya, serta bertukar informasi dalam ekosistem P2P (Oliveira dkk. 2017).

II.5 *Trust-Free System*

Trust-free system adalah konsep yang bertujuan untuk menghilangkan kebutuhan akan kepercayaan terhadap pihak ketiga maupun antar pengguna dalam sistem *peer-to-peer*, khususnya melalui penerapan teknologi *blockchain*. *Blockchain* memungkinkan pencatatan transaksi yang bersifat *immutable*, disepakati secara konsensus, dan tersedia secara publik, sehingga mengurangi risiko yang timbul dari kepercayaan interpersonal atau kepada lembaga perantara (Gan dan Lau 2024). Dengan demikian, dalam *trust-free system*, kepercayaan tidak lagi diberikan kepada individu atau institusi, melainkan dialihkan ke dalam kode dan protokol yang bersifat transparan dan tidak dapat diubah.

Namun, berdasarkan studi yang telah dilakukan oleh Hawlitschek, Notheisen, dan Teubner (2018), terdapat sejumlah tantangan untuk mencapai sistem yang benar-benar *trust-free* dalam *blockchain*, yaitu:

1. Beberapa *smart contract* yang di-*deploy* pada platform seperti Ethereum belum sepenuhnya memenuhi kriteria *trust-free* karena kerentanan dalam desain atau kode.
2. *Blockchain* hanya menggeser kepercayaan dari otoritas pusat menjadi kepercayaan terhadap algoritma berserta orang-orang yang menciptakan dan memeliharanya (*algorithmic trust*).
3. Keberhasilan sistem ini juga bergantung pada faktor *sociotechnical* (GeeksforGeeks 2022), seperti kepercayaan terhadap layanan pihak ketiga dan legitimasi teknologi dalam kerangka hukum yang berlaku.

Agar *trust-free system* dapat diimplementasikan secara optimal, *smart contract* harus dirancang dengan keandalan tinggi, transparan, mematuhi regulasi, dan berfungsi sebagaimana mestinya. Keandalan ini mencakup kemampuan *smart contract* untuk mengeksekusi perintah secara otomatis tanpa campur tangan pihak ketiga, serta mencegah terjadinya manipulasi atau kesalahan dalam proses transaksi. Dengan adanya desain *smart contract* yang kuat dan terpercaya,

pengguna akan merasa lebih aman dan yakin untuk berinteraksi dalam sistem yang dibangun di atas teknologi *blockchain* (Gan dan Lau 2024).

II.6 *Zero-Knowledge Proof* (ZKProof)

ZKProof merupakan konsep yang pertama kali diperkenalkan oleh Goldwasser, Micali, dan Rackoff (1985) sebagai solusi untuk masalah privasi dalam interaksi digital, yang memungkinkan seseorang (*prover*) untuk membuktikan kepada pihak lain (*verifier*) bahwa suatu pernyataan benar tanpa mengungkapkan informasi lain selain kebenaran pernyataan tersebut (Dieye dkk. 2023). Hal ini dicapai dengan meminta pembuktian menghasilkan bukti yang memenuhi serangkaian kriteria tertentu, yang kemudian dapat digunakan oleh pemverifikasi untuk memverifikasi klaim tanpa mempelajari apa pun tentang pernyataan tersebut (Moya dkk. 2023).

ZKProof terbagi menjadi dua kategori utama, yaitu *Interactive Zero-Knowledge Proof* (IZKP) dan *Non-Interactive Zero-Knowledge Proof* (NIZKP). IZKP membutuhkan komunikasi berulang antara *prover* dan *verifier*. Contoh dari IZKP adalah protokol klasik dari Goldwasser-Micali-Rackoff. NIZKP memungkinkan pembuktian tanpa komunikasi langsung, sering kali menggunakan asumsi tertentu seperti fungsi *hash* yang aman secara kriptografi (Dieye dkk. 2023). Salah satu implementasi populer dari NIZKP adalah zk-SNARK (*Zero-Knowledge Succinct Non-Interactive Argument of Knowledge*), yang memungkinkan pembuktian dilakukan secara ringkas dan efisien (Chen dkk. 2022). zk-SNARK memiliki beberapa sifat utama:

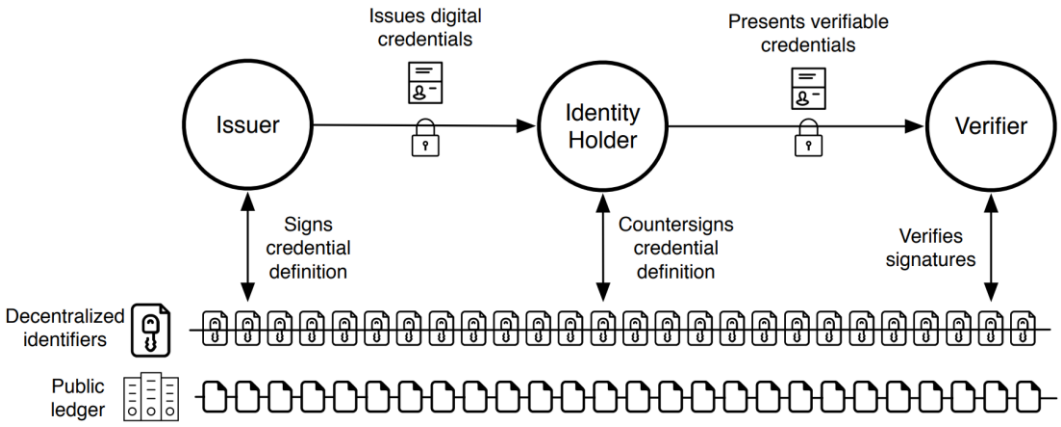
a. *succinct* (ukuran *proof* kecil dan waktu verifikasi cepat),
b. *non-interactive* (tidak perlu komunikasi berulang antara prover dan verifier), dan
c. *argument of knowledge* (pembuktian hanya bisa dilakukan jika prover benar-benar mengetahui data yang relevan).

II.7 *Self-Sovereign Identities* (SSI)

SSI merupakan paradigma dalam manajemen identitas digital, yang memungkinkan individu memiliki kendali penuh atas data identitas mereka tanpa bergantung pada pihak ketiga untuk mengelola atau memverifikasi identitas tersebut (Raipurkar dkk. 2023). Paradigma ini muncul sebagai respons terhadap kelemahan model *centralized identity*, seperti risiko pelanggaran privasi, penyalahgunaan data, dan ketergantungan pada otoritas pusat. Meskipun belum ada definisi yang tepat mengenai apa yang dimaksud dengan SSI, ada beberapa kriteria yang ditetapkan sebagai prinsip-prinsip yang mendasari SSI (Wang dan De Filippi 2019). Implementasi penuh SSI berdasarkan kriteria tersebut masih dalam tahap pengembangan dan eksperimen. Meskipun demikian, konsep ini diharapkan dapat merubah cara kita mengelola dan membuktikan identitas di dunia digital, menjadikan sistem lebih aman, transparan, dan *user-centric*.

II.7.1 *Verifiable Credentials* (VC)

VC adalah representasi digital dari klaim atau pernyataan yang dikeluarkan oleh pihak terpercaya (*issuer*) kepada pemiliknya (*holder*). VC memungkinkan pemilik untuk membuktikan kebenaran klaim tersebut kepada pihak lain (*verifier*) tanpa perlu bergantung pada otoritas pusat (Sporny, Longley, dan Chadwick 2022), seperti yang ditunjukkan pada Gambar II.3. Menurut World Wide Web Consortium (W3C), VC dirancang untuk mendukung privasi pengguna dengan fitur seperti *selective disclosure* dan ZKProof. Sebagai contoh, seorang *guest* ingin membuktikan bahwa dirinya berumur 18 tahun ke atas saat melakukan *hotel booking*. Dengan fitur *selective disclosure* pada VC, *guest* tersebut hanya perlu men-*generate proof* bahwa dirinya berumur di atas 18 tahun. Bahkan, dengan adanya ZKProof *guest* tersebut tidak perlu menunjukkan data tanggal lahirnya.



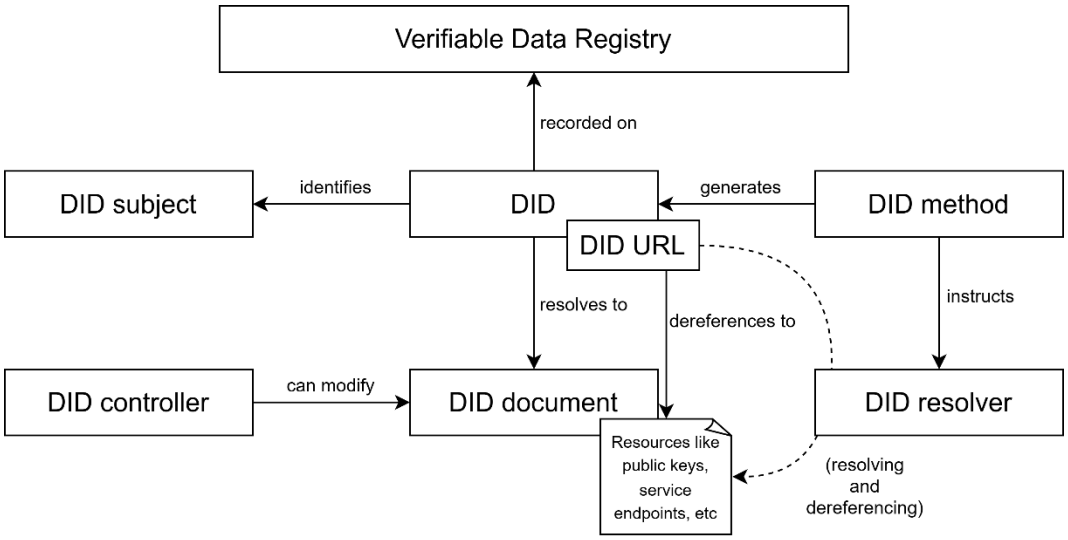
Gambar II.3 Cara kerja sistem VC (Lux dkk. 2020)

Agar *credential* dapat diverifikasi secara independen, sistem VC memanfaatkan *verifiable data registry*, berupa *blockchain* atau infrastruktur terdesentralisasi lainnya, yang menyimpan informasi metadata seperti Decentralized Identifiers (DIDs) dan *public key* milik *issuer*. Komponen ini memungkinkan *verifier* untuk memastikan bahwa suatu *credential* benar-benar berasal dari *issuer* yang sah tanpa perlu kontak langsung dengan *issuer*. VC memungkinkan individu untuk mengelola klaim mereka secara mandiri, seperti bukti kelayakan finansial, sertifikat pendidikan, atau identitas pribadi. Pada P2P AP berbasis *blockchain*, VC dapat digunakan untuk verifikasi *booking credential* yang dimiliki *guest* atau *host* di platform tanpa membagikan data sensitif kepada *verifier*, sehingga keamanan dan privasi data terjaga. VC disimpan secara terdesentralisasi oleh pemiliknya dan dapat diverifikasi menggunakan *decentralized identifiers* (DIDs) dan teknologi *blockchain*.

II.7.2 *Decentralized Identifiers* (DIDs)

DIDs adalah standar identitas terdesentralisasi yang memungkinkan pengguna memiliki kontrol penuh atas identitas digital mereka tanpa ketergantungan pada otoritas pusat. DIDs dikelola melalui *blockchain* untuk memastikan transparansi dan keamanan. Gambar II.4 menunjukkan komponen-komponen yang ada pada arsitektur DIDs (Reed dkk. 2020). Secara garis besar, arsitektur DIDs terdiri dari beberapa komponen utama, yaitu:

a. DID, yaitu *decentralized identifier* unik yang mengacu pada subjek. DID memiliki sintaks `did:method:unique-identifier`;
b. DID *document*, yaitu dokumen yang berisi metadata terkait DID, berupa file JSON-LD;
c. DID *method*, yaitu spesifikasi/aturan khusus untuk mengatur cara DID dibuat, direkonstruksi, dihapus, dll.



Gambar II.4 Arsitektur DIDs (Reed dkk. 2020)

DIDs memungkinkan implementasi SSI yang sepenuhnya terdesentralisasi, sehingga pengguna memiliki kontrol penuh atas identitas dan data pribadi mereka. Pendekatan ini mendukung peningkatan privasi dan keamanan, sekaligus mengurangi ketergantungan pada otoritas pusat. Kombinasi DIDs dengan teknologi *blockchain* memperkuat interoperabilitas di berbagai platform, termasuk P2P AP berbasis *blockchain*.

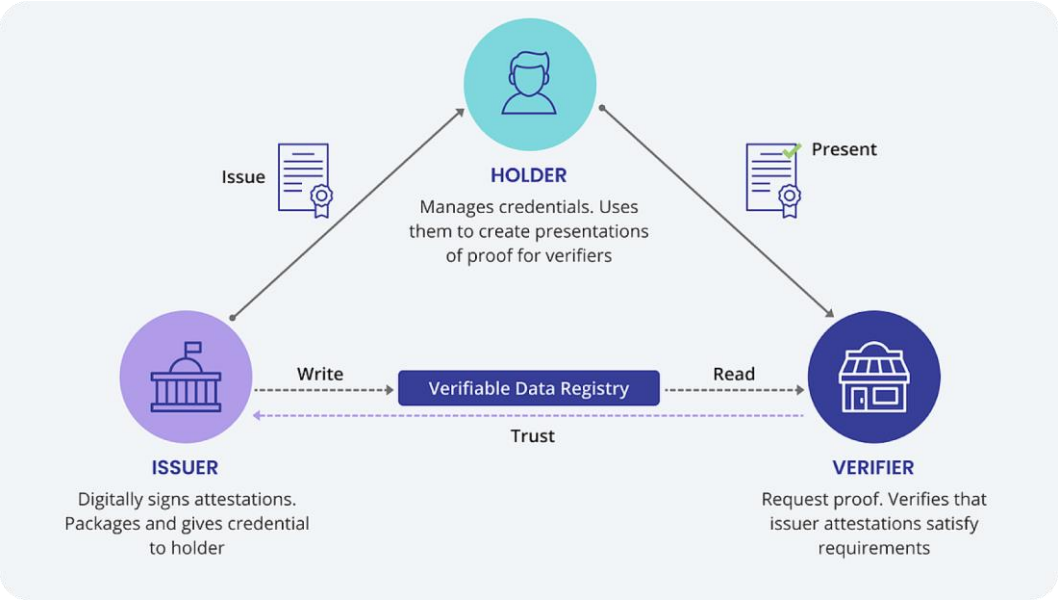
II.7.3 *Trust Triangle*

Model *trust triangle* merupakan kerangka fundamental dalam sistem identitas digital berbasis SSI, yang menggambarkan hubungan kepercayaan antara tiga entitas utama, yaitu *issuer* (penerbit), *holder* (pemegang), dan *verifier* (pemeriksa). Model ini menjadi dasar dalam implementasi arsitektur VC sebagaimana

didefinisikan oleh World Wide Web Consortium (Sporny, Longley, dan Chadwick 2022).

Dalam model ini, *issuer* bertindak sebagai pihak yang menerbitkan *verifiable credential* (VC) kepada holder. *Holder* menyimpan kredensial tersebut dalam *identity wallet* dan memiliki kendali penuh atas informasi tersebut. Ketika *verifier* membutuhkan bukti atas suatu klaim (misalnya status KYC, usia, atau keanggotaan), *holder* dapat menyusun dan menyerahkan *verifiable presentation* (VP) yang berisi satu atau lebih VC untuk diverifikasi.

Karakteristik utama dari *trust triangle* adalah *verifier* tidak perlu mengenal atau berinteraksi langsung dengan *issuer*. Verifikasi dilakukan secara kriptografis, yakni dengan memeriksa tanda tangan digital pada kredensial yang diterbitkan oleh *issuer*. Selama *verifier* mempercayai kredibilitas *issuer* yang menerbitkan VC, maka proses verifikasi tetap sah dan valid. Hal ini memungkinkan sistem identitas yang *trustless*, interoperabel, dan terdesentralisasi, tanpa bergantung pada otoritas identitas pusat.



Gambar II.5 Model *trust triangle* (cheqd 2023)

Model yang ditunjukkan pada Gambar II.5 membentuk *triangle of trust* sebagai berikut: (1) *issuer* memberikan VC kepada holder, (2) *holder* menyerahkan VP

kepada *verifier*, dan (3) *verifier* memverifikasi tanda tangan *issuer* untuk menilai keabsahan informasi. Dalam hal ini, kepercayaan tidak bergantung pada hubungan pribadi atau institusional antar aktor, melainkan pada kriptografi dan infrastruktur desentralisasi seperti DID dan jaringan *blockchain*.

Dengan demikian, *trust triangle* berperan penting dalam menciptakan ekosistem identitas digital yang aman, privat, dan dapat diverifikasi secara independen, serta mendukung prinsip *user-centric identity management* yang menjadi inti dari pendekatan SSI.

II.7.4 Iden3

Iden3 adalah protokol identitas digital terdesentralisasi berbasis *blockchain* yang dirancang untuk memungkinkan verifikasi identitas yang privat, aman, dan terkontrol oleh pengguna. Protokol ini dirancang agar kompatibel dengan *blockchain* berbasis EVM dan dibangun menggunakan *cryptographic primitives* seperti zk-SNARKs untuk menjamin privasi dan efisiensi proses verifikasi. Protokol ini mengimplementasikan VC dan DID sesuai dengan standar yang ditetapkan oleh W3C. Iden3 memiliki arsitektur berbasis zk-SNARK, yang memungkinkan pengguna menghasilkan *proof* yang dapat diverifikasi *on-chain* tanpa membuka data asli. Iden3 menggunakan BabyJubJub *elliptic curve* untuk mendukung efisiensi dalam *proof generation* dan *proof verification* dalam sirkuit *zero-knowledge* (Iden3 2023).

II.7.5 Privado ID

Privado ID adalah platform DID yang memungkinkan entitas, baik individu maupun organisasi, untuk mengelola dan membagikan bukti verifikasi mereka secara aman dan privat menggunakan teknologi SSI dan VCs. Privado ID dirancang untuk memungkinkan siapa pun membuktikan klaim tertentu tanpa harus mengungkapkan data pribadi secara langsung kepada *verifier* (Mittal 2025). Privado ID dibangun di atas protokol Iden3, sehingga setiap entitas dapat memiliki identitas digital yang dapat diverifikasi di berbagai platform yang kompatibel.

Proses verifikasi dilakukan melalui mekanisme ZKProofs, yang memungkinkan pengguna membuktikan validitas suatu klaim tanpa membocorkan isi dari klaim tersebut, seperti membuktikan bahwa mereka telah melakukan reservasi, tanpa mengungkapkan identitas atau detail reservasi tersebut.

Privado ID tidak menyimpan VC secara langsung di *blockchain*. Sebagai gantinya, *credential* disimpan *off-chain* dalam *identity wallet* yang dikelola oleh pemilik identitas. *Credential* ini disimpan di *cloud storage* yang dienkripsi, hanya dapat diakses oleh pemilik menggunakan *storage keys* mereka, sehingga menjaga keamanan dan privasi data (Mittal 2025). Sementara itu, *cryptographic proof*, seperti *merkle roots* atau ZKProofs, disimpan dan diverifikasi secara *on-chain* untuk memastikan keaslian *credential* tanpa mengungkapkan data sensitif. Privado ID memungkinkan pengembang untuk melakukan verifikasi baik secara *off-chain* maupun *on-chain*, sehingga dapat diintegrasikan secara fleksibel ke dalam sistem berbasis *blockchain* dan Web3 (Mittal 2025).

II.8 BabyJubJub Elliptic Curve

BabyJubJub adalah sebuah kurva eliptik yang dirancang untuk efisiensi dalam sistem ZKProof, khususnya pada protokol zk-SNARKs (Whitehat, Baylina, dan Bellés 2019). Kurva ini mendukung penggunaan algoritma tanda tangan seperti EdDSA (Basha dkk. 2021), yang sering digunakan dalam sistem identitas terdesentralisasi dan verifikasi seperti Privado ID atau zkKYC. BabyJubJub tidak cocok untuk digunakan di luar sirkuit *zero knowledge proof*, seperti di EVM langsung, karena tidak kompatibel dengan kurva standar EVM (seperti secp256k1), sehingga penggunaannya biasanya terbatas pada *off-chain proof generation* dan *on-chain verifier*. Pada penelitian ini, BabyJubJub digunakan sebagai bagian dari sistem verifikasi tanda tangan dan autentikasi pengguna secara kriptografis di dalam sirkuit ZKProof, baik untuk membuktikan status *booking*, identitas, maupun kepemilikan *credential* secara anonim dan efisien.

II.9 *Merkle Tree*

Merkle tree adalah struktur data berbentuk pohon biner yang digunakan untuk memastikan integritas dan konsistensi data dalam sistem terdistribusi. Setiap daun dari pohon berisi *hash* dari suatu data, dan setiap *node* lainnya merupakan *hash* dari penggabungan dua *node* anaknya (Dahlberg, Pulls, dan Peeters 2016). Hal ini memungkinkan proses verifikasi data yang efisien karena hanya memerlukan bukti dalam bentuk jalur hash dari data yang ingin diverifikasi ke *root tree* tersebut (*merkle root*). *Merkle tree* banyak digunakan dalam *blockchain* (seperti Ethereum) untuk memverifikasi transaksi tanpa harus memproses seluruh data, sehingga mengurangi beban komputasi dan *bandwidth*.

II.10 Sparse Merkle Tree (SMT)

Sparse merkle tree (SMT) merupakan varian dari *merkle tree* yang digunakan dalam sistem identitas berbasis *blockchain*, seperti yang diterapkan oleh Iden3. Berbeda dengan *merkle tree* konvensional, SMT memiliki struktur khusus, yaitu setiap elemen data berada pada posisi yang telah ditentukan berdasarkan indeksnya, dengan jumlah total 2^{256} daun. Struktur ini memungkinkan pembuktian efisien terhadap keberadaan (*proof of membership*) maupun ketidakhadiran (*proof of non-membership*) suatu elemen dalam pohon (Dahlberg, Pulls, dan Peeters 2016). Pada penelitian ini, SMT digunakan sebagai bagian dari mekanisme verifikasi status credential secara efisien dan privat. Dengan kombinasi zk-SNARK dan SMT, pengguna dapat memberikan bukti validasi identitas atau pemesanan (*booking*) secara anonim namun tetap dapat diverifikasi oleh sistem secara kriptografis.

II.11 Penelitian Terkait

Pada pelaksanaan penelitian ini, dilakukan kajian terhadap berbagai studi yang berkaitan dengan pengembangan P2P AP berbasis *blockchain*. Kajian ini bertujuan untuk memahami pendekatan, metode, serta temuan yang telah dilakukan oleh peneliti sebelumnya. Tabel II.1 menyajikan penjelasan singkat mengenai berbagai

penelitian terkait yang menjadi referensi dalam pelaksanaan penelitian, serta relevansinya terhadap penelitian.

Tabel II.1 Perbandingan penelitian terdahulu

No	Judul	Nama Penulis	Keterkaitan
1	*Blockchain and Trust in the Platform Economy: The Case of Peer-to-Peer Sharing*	David Dann, Christian Peukert, Carl Martin, Christof Weinhardt, dan Florian Hawlitschek.	Penelitian ini memberikan wawasan tentang cara teknologi *blockchain* dapat meningkatkan kepercayaan dalam platform ekonomi berbasis P2P, yang relevan untuk menciptakan transparansi dan keandalan dalam sistem P2P AP.
2	*A Self-Sovereign Identity Based on Zero-Knowledge Proof and Blockchain*	Mohameden Dieye, Pierre Valiorgue, Jean Patrick Gelas, El Hacen Diallo, Parisa Ghodous, Frederique Biennier, dan Eric Peyrol.	Penelitian ini memperkuat konsep penerapan SSI menggunakan ZKProof untuk memverifikasi data pengguna tanpa membocorkan informasi sensitif, yang merupakan komponen utama dalam desain P2P AP.
3	*Digital Identity System Using Blockchain-based Self Sovereign Identity & Zero Knowledge Proof*	Abhijeet R. Raipurkar, Shreyas Bobde, Anurag Tripahi, dan Mohit Sahu.	Penelitian ini relevan dalam implementasi SSI berbasis *blockchain* dan ZKProof, yang merupakan bagian integral dari P2P AP untuk melindungi privasi pengguna.

II.11.1 *Blockchain and Trust in the Platform Economy: The Case of Peer-to-Peer Sharing*

Penelitian ini membahas cara *blockchain* dapat meningkatkan kepercayaan dalam ekonomi berbasis platform, khususnya dalam konteks *peer-to-peer* (P2P) *sharing*. Penelitian menggarisbawahi pergeseran kepercayaan dari individu sebagai mitra transaksi ke kepercayaan pada platform secara keseluruhan, terutama dalam konteks teknologi *blockchain*. Meskipun *blockchain* sering disebut sebagai *trust-free systems*, kepercayaan terhadap teknologi *blockchain* itu sendiri ternyata

memiliki dampak signifikan terhadap kepercayaan pada platform yang menggunakannya. Oleh karena itu, platform perlu memastikan kepercayaan terhadap teknologi *blockchain* agar dapat berhasil diimplementasikan.

Bagi platform baru yang belum memiliki basis pengguna yang mapan, teknologi *blockchain* dapat membantu mengatasi masalah *cold start*, yaitu kondisi awal saat belum ada transaksi yang terjadi pada platform, yang juga menyebabkan sistem reputasi tradisional (seperti *star ratings* dan *review*) masih sangat terbatas. *Blockchain* dapat mendukung pengguna untuk memulai transaksi pertama mereka, yang pada akhirnya membantu membangun reputasi di platform. Meski demikian, sistem reputasi tradisional, seperti ulasan bintang atau teks, tetap diperlukan untuk membantu pengguna merasa percaya terhadap mitra berbagi mereka.

Bagi platform yang sudah mapan, *blockchain* juga memiliki manfaat, terutama ketika sistem reputasi tradisional mulai kehilangan nilai akibat penilaian positif yang berlebihan atau diskriminasi. Dengan menggunakan *blockchain* sebagai teknologi dasar, kepercayaan terhadap platform dan kelangsungan transaksinya dapat tetap terjaga. Kombinasi *blockchain* dan sistem reputasi tradisional dianggap sebagai strategi yang efektif untuk menjaga kepercayaan pengguna dalam jangka panjang.

II.11.2 *A Self-Sovereign Identity Based on Zero-Knowledge Proof and Blockchain*

Penelitian ini berfokus pada pengembangan sistem identitas digital terdesentralisasi yang bertujuan untuk meningkatkan keamanan dan privasi pengguna. Dalam prosesnya, penelitian ini mengeksplorasi konsep SSI secara mendalam, termasuk prinsip-prinsip dasar yang mendasarinya. Selain itu, penelitian ini juga menyoroti penerapan teknologi ZKProof dan *blockchain* sebagai fondasi utama untuk membangun sistem identitas yang aman, privat, dan bebas dari ketergantungan pada otoritas pusat.

Pendekatan yang diusulkan dalam penelitian ini adalah menggunakan protokol ZKProof dan teknologi *blockchain* untuk memungkinkan pengguna mengelola identitas digital mereka secara mandiri. Pengguna tidak perlu bergantung pada pihak ketiga atau otoritas terpusat untuk memverifikasi identitas mereka, yang menjaga privasi dan meminimalkan pengungkapan informasi pribadi. Selain itu, protokol SSI yang diajukan mematuhi standar regulasi seperti eIDAS (European Commission 2025) dan GDPR (Intersoft Consulting, t.t.), sehingga verifikasi identitas dapat dilakukan dengan aman tanpa mengorbankan privasi. Penelitian ini juga membahas implementasi teknis protokol SSI dan kemungkinan penerapannya di berbagai sektor yang membutuhkan verifikasi identitas yang aman dan terdesentralisasi.

II.11.3 *Digital Identity System Using Blockchain-based Self Sovereign Identity & Zero Knowledge Proof*

Penelitian ini menekankan pada penciptaan solusi identitas digital berbasis *blockchain* dalam konteks ketidakpercayaan antara pihak-pihak yang terlibat. Solusi ini dirancang untuk menghilangkan kebutuhan akan perantara tepercaya dengan memanfaatkan mekanisme verifikasi terdesentralisasi. Berbeda dengan sistem identitas tradisional yang sering bergantung pada penyimpanan terpusat, penelitian ini mengusulkan penggunaan SSI untuk memberikan pengguna kontrol penuh atas identitas mereka secara lebih terdesentralisasi dan aman.

Pendekatan yang diusulkan dalam penelitian ini menciptakan sistem identitas yang sepenuhnya terdesentralisasi, yaitu identitas asli pengguna disimpan dalam aplikasi web mereka masing-masing menggunakan penyimpanan terdesentralisasi. Proses validasi informasi identitas dilakukan melalui penggunaan protokol ZKProof untuk memastikan bahwa informasi yang diberikan hanya diketahui oleh pihak yang berwenang, tanpa perlu mengungkapkan detail pribadi pengguna. Solusi ini berfungsi sebagai dompet identitas digital, memungkinkan pengguna untuk memverifikasi identitas mereka dengan menggunakan teknologi *blockchain* dan pendekatan berbasis SSI. Sistem yang diusulkan ini juga mengatasi masalah yang

terkait dengan keabadian data, keterlacakan, dan kontrol terpusat yang sering ditemukan pada sistem identifikasi tradisional. Selain itu, sistem ini menawarkan model klaim yang dapat diverifikasi oleh pihak eksternal, memberikan tingkat keamanan dan kepercayaan yang lebih tinggi dalam sistem identitas digital.

BAB III

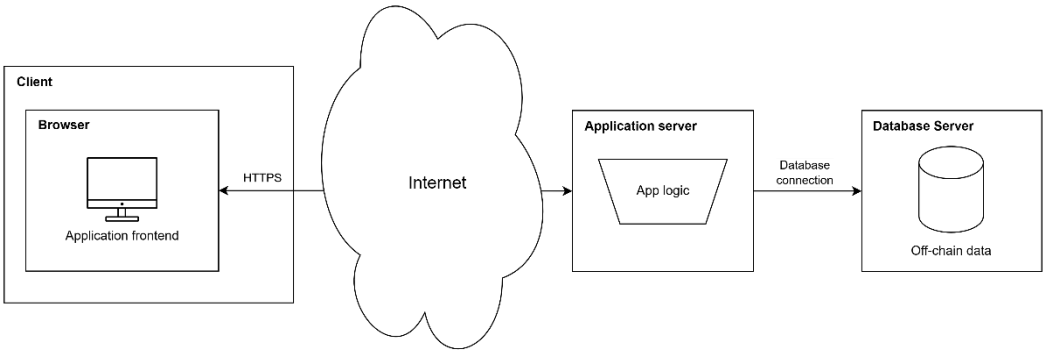
ANALISIS MASALAH

III.1 Analisis Kondisi Saat Ini

Berikut adalah hasil identifikasi masalah yang ada dalam P2P AP tradisional. Fakta-fakta dikumpulkan melalui eksplorasi informasi latar belakang, mengidentifikasi masalah utama yang ada dalam sistem tradisional, serta mengumpulkan data dari berbagai sumber, seperti buku, *conference paper*, berita, dan jurnal. Tahapan ini bertujuan untuk memberikan pemahaman yang mendalam mengenai konteks masalah yang menjadi fokus penyelesaian.

III.1.1 Masalah Keamanan dan Privasi pada *Centralized Architecture*

Pada *centralized architecture*, data disimpan dan dikelola di satu *server* utama atau lokasi pusat, seperti pada Gambar III.1, sehingga dapat diakses dan diperbarui dengan mudah (Raj, 2019). Pengelolaan seluruh data dan informasi dilakukan oleh satu entitas, biasanya penyedia layanan penyimpanan. Data pengguna diunggah dan disimpan di *server* yang sepenuhnya berada di bawah kendali penyedia. Pendekatan ini memberikan kendali penuh kepada penyedia atas berbagai aspek manajemen data, termasuk keamanan.



Gambar III.1 Gambaran umum *centralized architecture*

Pendekatan ini memberikan manfaat seperti kemudahan pengelolaan data, konsistensi informasi, dan efisiensi dalam pengawasan serta pemeliharaan sistem. Dengan kendali yang terpusat, penyedia dapat memastikan data dikelola sesuai standar keamanan dan kebijakan yang ditetapkan, sehingga meminimalkan risiko kehilangan data akibat kesalahan pengguna. Namun demikian, pendekatan ini juga memungkinkan sejumlah masalah terkait keamanan dan privasi (Rosoon, Choksuchat, dan Aiyarak 2023), terutama pada konteks P2P AP, yang tentunya dapat mengurangi kepercayaan pengguna.

III.1.1.1 *Censorship*

Kontrol yang terpusat dapat membuka peluang terjadinya sensor terhadap data pengguna, baik secara disengaja maupun tidak. Penyedia P2P AP dapat memutuskan untuk melarang pengguna tertentu atau jenis properti tertentu tanpa transparansi, dan pemilik properti bisa kehilangan akses ke platform tanpa pemberitahuan atau alasan yang jelas. *Censorship* juga dapat berupa penghapusan ulasan secara sepihak, seperti yang pernah dilakukan oleh AirBNB (Blengini dan Venturini 2024).

III.1.1.2 Serangan Siber pada *Server* Pusat

Salah satu masalah keamanan dan privasi yang sering muncul pada *centralized architecture* adalah *single point of failure*. Pada sistem ini, semua data dan proses disimpan serta dikelola di satu *server* pusat, yang menjadi target utama serangan siber. Jika *server* tersebut disusupi atau mengalami kegagalan, seluruh layanan akan terhenti, dan data pengguna bisa terekspos atau hilang. Selain itu, dengan terpusatnya data, risiko pencurian atau manipulasi informasi menjadi lebih tinggi karena peretas hanya perlu fokus pada satu titik untuk mendapatkan akses ke seluruh sistem. Hal ini membuat arsitektur terpusat lebih rentan terhadap serangan seperti *Distributed Denial of Service* (DDoS), metode *cyber attack* dengan membanjiri *server* dengan *traffic* berlebih sehingga sistem tidak dapat beroperasi

secara normal. Keterbatasan ini menimbulkan kekhawatiran besar terhadap keamanan dan privasi pengguna, terutama ketika data yang disimpan bersifat sensitif atau rahasia.

III.1.1.3 Monopoli Kendali oleh Penyedia P2P AP

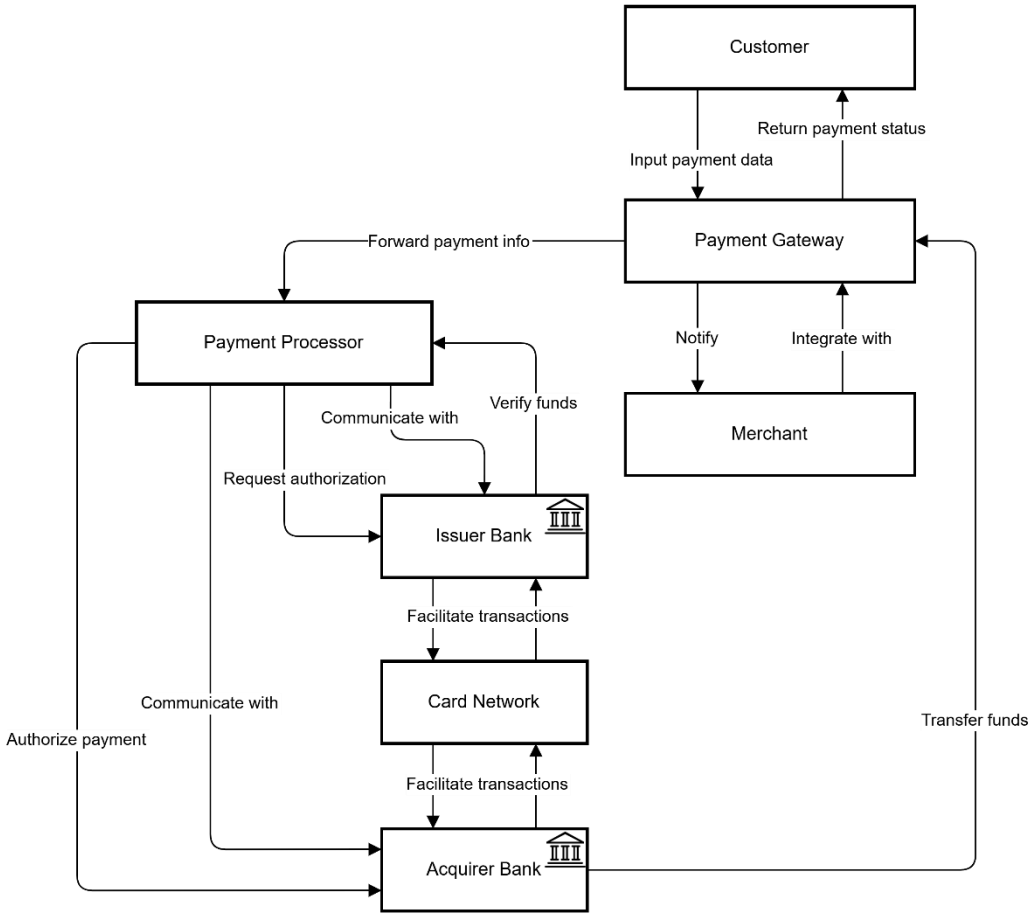
Ketergantungan pada satu penyedia layanan dapat menimbulkan kekhawatiran terkait *centralized control*, karena penyedia P2P AP memiliki wewenang penuh untuk mengatur akses dan penggunaan data tanpa melibatkan pengguna secara langsung (Lux dkk. 2020). Hal ini menciptakan ketidakseimbangan kekuasaan antara platform dan pengguna, yang berpotensi merugikan pihak pengguna. Situasi seperti ini dapat menimbulkan ketidakpuasan, terutama jika terjadi pelanggaran privasi atau penyalahgunaan data oleh pihak penyedia.

III.1.2 *Third-Party Risk* dalam Proses Transaksi

Di dunia keuangan, keterlibatan pihak ketiga dalam proses transaksi dan penyediaan layanan digital semakin umum, terutama untuk platform P2P. Namun, hal ini membawa risiko signifikan, terutama terkait dengan keamanan data dan ketergantungan pada penyedia layanan eksternal yang tidak selalu memiliki kontrol yang cukup terhadap potensi ancaman siber. Hal ini menjadi semakin relevan ketika kita melihat cara sebuah sistem transaksi digital bekerja dengan melibatkan pihak ketiga, seperti yang tergambar pada Gambar III.2. Alur transaksi tidak hanya melibatkan pembeli dan penjual, tetapi juga melalui beberapa entitas eksternal seperti *payment gateway*, *payment processor*, bank penerbit, dan bank penerima. Setiap titik interaksi ini, terutama yang melibatkan pertukaran data finansial dan identitas pribadi, merupakan potensi serangan apabila tidak dikelola dengan sistem keamanan yang ketat. Misalnya, jika *payment gateway* atau *processor* disusupi, maka data sensitif pengguna dan dana transaksi bisa berisiko bocor atau disalahgunakan.

Salah satu contoh nyata terkait risiko pihak ketiga adalah kebocoran data yang terjadi pada Finastra, perusahaan teknologi finansial yang menyediakan layanan

kepada sejumlah besar bank besar. Wall Street Journal melaporkan bahwa Finastra mengalami pelanggaran data yang disebabkan oleh akses tidak sah ke platform transfer file yang aman (Rundle 2024). Kejadian ini memperlihatkan cara pihak ketiga yang bertanggung jawab atas pengelolaan data dan sistem dapat menjadi titik lemah dalam ekosistem keamanan. Meskipun sistem internal Finastra aman, pihak ketiga yang memiliki akses dapat menjadi saluran bagi serangan siber yang mengakibatkan kebocoran data yang besar.



Gambar III.2 Sistem transaksi menggunakan *third party provider*

Ketergantungan pada penyedia layanan pihak ketiga juga menambah kompleksitas dalam hal pengelolaan risiko. Sebagai contoh, Fidelity Investments baru-baru ini melaporkan adanya pelanggaran data yang melibatkan akses tidak sah pada dua akun pelanggan yang baru dibuka. Berdasarkan laporan dari The Sun, pelanggaran

ini menyebabkan kebocoran informasi pribadi sensitif, termasuk nomor Jaminan Sosial dan SIM (Malcolm 2024). Hal ini menunjukkan cara pihak ketiga yang terlibat dalam pengelolaan akun atau data dapat menjadi titik rawan yang menambah risiko bagi perusahaan yang bergantung pada mereka.

Penting juga untuk memperhatikan cara ketergantungan pada teknologi yang dikelola oleh pihak ketiga dapat memengaruhi operasi suatu perusahaan. The Times melaporkan bahwa salah satu regulator finansial di United Kingdom, Financial Conduct Authority (FCA), menekankan pentingnya kesiapan sektor keuangan untuk menghadapi krisis teknologi, seperti yang terjadi akibat gangguan layanan yang disebabkan oleh perusahaan pihak ketiga seperti CrowdStrike (*The Times* 2024). Pada kasus ini, gangguan yang memengaruhi jutaan perangkat di seluruh dunia menggarisbawahi betapa pentingnya perusahaan keuangan untuk mengelola risiko yang timbul dari penyedia layanan pihak ketiga, terutama terkait dengan potensi kerugian akibat serangan siber.

III.1.3 Kurangnya Keseimbangan antara Transparansi dan Privasi

Kepercayaan merupakan elemen yang sangat penting dalam P2P AP karena memengaruhi keputusan pemesanan pengguna. Tanpa adanya kepercayaan, pengguna cenderung ragu untuk berinteraksi atau melakukan transaksi di dalam platform. Dari perspektif sosial, kepercayaan dibangun di atas harapan bahwa semua pihak yang terlibat dalam sistem layanan, seperti layanan P2P AP, akan bertindak sesuai aturan (Agag dan Eid 2019; Tussyadiah dan Park 2018).

Identitas merupakan salah satu komponen penting dalam P2P AP untuk membangun kepercayaan antara *host* dengan *guest*. Ketika satu pihak mempercayai pihak lainnya, pihak tersebut memiliki ekspektasi bahwa pihak lain tersebut akan bertindak sesuai dengan cara dia mempresentasikan identitasnya dalam P2P AP. Hal ini berlaku untuk *guest*, *host*, maupun pihak-pihak lain yang terlibat dalam P2P AP.

Identitas ini memungkinkan *guest* dan *host* untuk membuat keputusan yang lebih informasional sebelum melakukan transaksi. Namun, terlalu banyak persyaratan identitas dapat menimbulkan resistansi. Pengguna internet seringkali enggan untuk memberikan informasi yang sensitif tentang dirinya (Agag dan Eid 2019). Hal ini menjadi tantangan besar bagi P2P AP untuk menciptakan keseimbangan antara kebutuhan akan transparansi identitas dan perlindungan privasi pengguna. Penyedia platform harus mampu menyediakan mekanisme verifikasi identitas yang andal untuk meningkatkan kepercayaan, namun di sisi lain tetap menghormati preferensi privasi pengguna.

III.1.4 Keterbatasan Sistem Reputasi Tradisional

Untuk membangun kepercayaan antar *peers*, P2P tradisional menggunakan *star ratings*, ulasan, dan foto profil sebagai indikator reputasi seorang pengguna. Namun, pendekatan ini memiliki kelemahan signifikan. *Star ratings* dan ulasan dapat dipengaruhi oleh bias subjektif (Veldhuizen 2020), baik yang disengaja maupun tidak disengaja. Beberapa ulasan tidak sepenuhnya merefleksikan kualitas layanan atau pengalaman sebenarnya, melainkan dipengaruhi oleh prasangka pribadi (Huang 2021), perbedaan budaya, balas dendam, atau motif tertentu seperti *tainted prosociality*, yaitu tindakan positif yang dilakukan dengan tujuan tersembunyi untuk keuntungan pribadi (Zloteanu dkk. 2021).

Masalah lain yang menjadi keterbatasan sistem reputasi tradisional adalah foto profil sebagai indikator reputasi. Foto profil dapat menimbulkan diskriminasi berbasis gender, ras, atau usia, yang berkontribusi pada ketidakadilan dalam P2P AP (Tushev, Ebrahimi, dan Mahmoud 2022). Selain itu, sistem reputasi tradisional sangat rentan terhadap manipulasi, seperti kasus penghapusan ulasan negatif pada platform AirBNB (Blengini dan Venturini 2024). Praktik manipulasi ini mengurangi efektivitas mekanisme reputasi dan menurunkan tingkat kepercayaan pengguna.

Semua keterbatasan ini menunjukkan perlunya mekanisme reputasi yang lebih netral, aman, dan objektif untuk membangun kepercayaan yang autentik dalam

ekosistem P2P. Mekanisme semacam ini harus mampu memberikan penilaian yang adil terhadap perilaku pengguna tanpa dipengaruhi oleh bias, kepentingan tertentu, atau manipulasi data. Dengan demikian, sistem reputasi yang dirancang secara transparan dan terdesentralisasi dapat menjadi fondasi penting dalam menciptakan lingkungan P2P yang lebih terpercaya dan berkelanjutan.

III.2 Analisis Kebutuhan

Tahap analisis kebutuhan merupakan fondasi utama untuk memastikan sistem yang dibangun mampu memenuhi tujuan fungsional dan non-fungsional secara optimal. Analisis kebutuhan harus mencakup pemetaan menyeluruh terhadap kebutuhan pengguna serta proses bisnis yang mendasari. Subbab ini akan menguraikan secara sistematis masalah pengguna, kebutuhan fungsional, serta kebutuhan non-fungsional yang akan digunakan sebagai dasar dalam perancangan dan implementasi sistem.

III.2.1 Identifikasi Masalah Pengguna

Analisis ini disusun berdasarkan studi literatur, berita, serta wawancara kualitatif dengan beberapa pengguna yang pernah menggunakan P2P AP. Tujuan dari analisis ini adalah mengidentifikasi permasalahan nyata yang dihadapi oleh masing-masing pemangku kepentingan dalam ekosistem P2P *accommodation*, sehingga solusi teknologi yang dikembangkan dapat benar-benar menjawab kebutuhan pengguna. Platform ini melibatkan tiga jenis pengguna utama: penyewa (*guest*), pemilik properti (*host*), dan pengelola platform. Setiap jenis pengguna memiliki permasalahan yang berbeda dan spesifik.

Wawancara kualitatif dilakukan secara langsung terhadap dua kelompok pengguna utama, yaitu penyewa (*guest*) dan pemilik properti (*host*), yang menjadi sumber data primer dalam studi ini. Selain itu, wawancara juga dilakukan terhadap penyedia platform P2P AP. Namun, data dari penyedia platform digunakan secara terbatas sebagai pelengkap dan tidak menjadi fokus utama dalam proses identifikasi masalah pengguna.

Dalam pelaksanaan wawancara, pendekatan yang digunakan disesuaikan dengan jenis pertanyaan yang diajukan kepada narasumber. Untuk pertanyaan-pertanyaan yang bersifat retrospektif, seperti pengalaman dan permasalahan saat menggunakan platform P2P AP yang sudah ada, narasumber diminta untuk menjawab berdasarkan pengalaman nyata mereka menggunakan platform yang dikenal luas, seperti Airbnb atau Couchsurfing.

Namun, untuk pertanyaan yang berkaitan dengan aspek kepercayaan (*trust*), seperti mengenai sikap pengguna terhadap identitas penyewa/pemilik, jaminan layanan, atau kredibilitas sistem, narasumber terlebih dahulu diberikan prekondisi simulatif. Dalam simulasi tersebut, mereka diminta membayangkan skenario di mana mereka menggunakan sebuah platform baru yang belum dikenal luas dan belum memiliki reputasi atau rekam jejak yang bisa dijadikan acuan. Penekanan diberikan bahwa konteks ini tidak merujuk pada platform populer, melainkan sebuah sistem baru tanpa elemen *trust* yang telah terbentuk sebelumnya, baik terhadap platform maupun terhadap sesama pengguna.

Pendekatan ini bertujuan untuk menggali kebutuhan dan kekhawatiran mendasar yang muncul ketika elemen kepercayaan belum terbentuk, agar analisis kebutuhan yang dihasilkan dapat disusun dalam kerangka yang lebih netral. Dengan demikian, fokus dapat diarahkan pada fitur-fitur fungsional dan struktur sistem yang diperlukan untuk membangun rasa aman dan kenyamanan, tanpa asumsi dukungan dari reputasi platform atau pengalaman pengguna sebelumnya.

III.2.1.1 Penyewa (*Guest*)

Penyewa (*guest*) adalah pengguna yang mencari dan memesan akomodasi melalui platform. Masalah yang sering mereka hadapi antara lain:

a. Beberapa penyewa merasa ragu saat memesan karena tidak yakin apakah tempat yang ditawarkan benar-benar sesuai dengan informasi yang ditampilkan. Informasi tentang pemilik properti sering kali terbatas atau kurang jelas.

b. Sebelum melakukan pemesanan, penyewa harus memberikan banyak informasi pribadi, tetapi mereka tidak mengetahui pihak-pihak yang bisa mengakses data tersebut dan tujuannya.
c. Sebagian pengguna kesulitan melakukan pembayaran karena hanya tersedia metode tertentu. Selain itu, proses pembayaran bisa memakan waktu atau menimbulkan kekhawatiran, terutama jika pemesanan dibatalkan sepihak.
d. Tamu mengandalkan ulasan pengguna lain untuk membuat keputusan, tetapi sering kali muncul kasus ulasan buruk dihapus atau disembunyikan oleh platform.
e. Beberapa tamu mengaku pernah tidak diterima karena foto atau nama mereka, yang menimbulkan rasa tidak adil dan kurangnya kenyamanan saat menggunakan platform.

III.2.1.2 Pemilik Properti (*Host*)

Pemilik properti (*host*) adalah pengguna yang menyewakan tempat tinggalnya melalui platform. Masalah yang mereka hadapi di antaranya:

a. Banyak pemilik properti khawatir bahwa tamu yang menginap akan merusak barang atau melanggar aturan rumah. Platform tidak selalu memberikan perlindungan atau ganti rugi yang jelas.
b. Pemilik properti tidak selalu mendapatkan informasi yang cukup tentang penyewa sebelum menerima pemesanan. Ini membuat mereka khawatir, terutama jika tamu baru pertama kali menggunakan platform.
c. Ada pemilik properti yang mengeluh bahwa pembayaran dari tamu baru diterima setelah proses yang panjang, sehingga menyulitkan pengelolaan keuangan mereka.
d. Jika platform menutup akun mereka secara sepihak atau tiba-tiba mengubah aturan, tuan rumah bisa kehilangan akses untuk menyewakan properti tanpa penjelasan yang adil.
e. Pemilik properti keberatan memberikan banyak informasi pribadi kepada platform.

III.2.1.3 Pengelola Platform

Pengelola platform bertanggung jawab atas jalannya sistem dan hubungan antara tamu dan tuan rumah. Mereka juga menghadapi sejumlah tantangan, seperti:

a. Karena semua informasi disimpan dan diatur oleh pihak platform, maka tanggung jawab terhadap kebocoran data atau kesalahan sangat besar dan bisa merugikan banyak pengguna.
b. Pengelola kadang dianggap lebih memihak salah satu pihak, terutama jika ulasan dihapus atau aturan berubah mendadak tanpa pemberitahuan yang jelas kepada pengguna.
c. Saat sistem ulasan bisa diatur, akun bisa dihapus sepihak, atau tidak ada jaminan perlindungan yang jelas, pengguna mulai meragukan integritas dan niat baik platform.
d. Saat terjadi kebocoran data pada *third party* yang digunakan oleh sistem, data pengguna platform ikut bocor, sehingga platform mengalami kerugian secara finansial.

III.2.2 Kebutuhan Fungsional

Kebutuhan fungsional memainkan peran penting sebagai landasan untuk merancang solusi teknologi yang tepat dengan mendefinisikan fitur-fitur utama yang harus dimiliki sistem agar dapat memenuhi tujuan utamanya, yaitu menciptakan *trust-free system* pada P2P AP. Kebutuhan-kebutuhan ini disusun berdasarkan masalah-masalah yang diidentifikasi dari ketiga kelompok pengguna, yaitu penyewa (*guest*), pemilik properti (*host*), dan pengelola platform. Setiap kebutuhan mencakup deskripsi mendetail mengenai fungsi yang harus dijalankan sistem. Tabel III.1 menyajikan daftar kebutuhan fungsional secara spesifik untuk setiap kelompok pengguna.

III.2.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah spesifikasi yang menetapkan batasan pada aspek-aspek sistem di luar layanan atau fungsi utamanya. Pemenuhan kebutuhan ini

memiliki dampak signifikan terhadap kinerja keseluruhan sistem. Kebutuhan non-fungsional mencakup aspek-aspek yang terdapat pada Tabel III.2.

Tabel III.1 Daftar kebutuhan fungsional

Kode	Kebutuhan
FR-01	Penyewa harus dapat melakukan pembayaran melalui dompet digital, tanpa perlu mengungkap informasi sensitif seperti nomor rekening atau identitas pribadi.
FR-02	Penyewa harus diberikan akses untuk mengajukan sengketa, sebagai bentuk perlindungan apabila layanan tidak sesuai dengan yang dijanjikan.
FR-03	Sistem perlu memberikan tanda bukti pemesanan yang hanya dapat diakses dan digunakan oleh penyewa yang bersangkutan, untuk menjamin eksklusivitas dan mencegah penyalahgunaan bukti tersebut oleh pihak lain.
FR-04	Pemilik properti harus dapat memverifikasi secara sah bahwa seseorang telah melakukan pemesanan.
FR-05	Sistem harus memungkinkan pemilik properti menerima jaminan keamanan (*host stake*) dari penyewa sebagai bentuk perlindungan terhadap risiko kerusakan atau pelanggaran aturan selama masa inap.
FR-06	Pemilik properti harus memiliki akses untuk mengajukan sengketa jika terjadi pelanggaran atau permasalahan selama proses pemesanan berlangsung.
FR-07	Sistem harus menyediakan bukti pemesanan yang tidak dapat dipalsukan, guna memastikan bahwa hanya pengguna yang sah yang dapat mengakses properti yang telah dipesan.
FR-08	Harus tersedia mekanisme penyelesaian sengketa (*dispute resolution*) yang adil dan transparan antara pemilik properti dan penyewa dalam kasus terjadi ketidaksesuaian atau perselisihan.
FR-09	Sistem perlu mengimplementasikan otomatisasi dalam pengelolaan dana, dengan menahan pembayaran sementara hingga proses pemesanan selesai sesuai ketentuan.
FR-10	Platform harus memiliki mekanisme pencegahan terhadap penyalahgunaan sistem, seperti pengguna yang mencoba membuat banyak akun palsu untuk mendapatkan keuntungan tidak sah.
FR-11	Diperlukan sistem jaminan transaksi bagi seluruh pihak (pemilik dan penyewa), untuk meningkatkan rasa aman dan akuntabilitas.
FR-12	Platform wajib memiliki proses verifikasi pengguna yang efektif, guna memastikan bahwa pengguna adalah individu yang sah, tanpa mengorbankan privasi mereka.
FR-13	Platform harus menyediakan sistem evaluasi untuk pemilik properti yang sangat sulit untuk dimanipulasi.

Tabel III.2 Daftar kebutuhan non-fungsional

Kode	Parameter	Penjelasan
NFR-1	*Auditability*	Setiap transaksi penting harus dapat ditelusuri secara historis oleh pengguna yang berwenang, tanpa mengungkap informasi pribadi pengguna lain.
NFR-2	Portabilitas	Aplikasi harus dapat digunakan di berbagai jenis perangkat dan sistem operasi (*mobile*, *tablet*, *desktop*) tanpa kehilangan fungsionalitas utama.
NFR-3	Waktu respon	Sistem harus mampu memberikan respons terhadap tindakan pengguna dalam waktu maksimal 10 detik untuk menjaga kenyamanan pengguna.

III.3 Analisis Pemilihan Solusi

Setelah menganalisis masalah, langkah selanjutnya adalah melakukan analisis terhadap berbagai solusi yang memungkinkan untuk mencapai tujuan penelitian. Analisis ini mencakup evaluasi pendekatan teknis dan konseptual yang paling sesuai dengan kebutuhan sistem dan kriteria desain. Tujuan utama dari penelitian ini adalah merancang dan mengimplementasikan *trust-free system* pada P2P AP guna mengurangi ketergantungan pada pihak ketiga dan meningkatkan kepercayaan melalui mekanisme terdesentralisasi.

III.3.1 Alternatif Solusi

Untuk mencapai tujuan utama penelitian, terdapat beberapa alternatif solusi yang dapat dipertimbangkan untuk mengatasi masalah yang telah diidentifikasi sebelumnya. Setiap alternatif memiliki pendekatan tersendiri yang perlu dianalisis secara mendalam sebelum menentukan solusi terbaik. Subbab ini akan menjelaskan berbagai alternatif solusi yang sudah ditampilkan pada Tabel III.3.

Tabel III.3 Daftar alternatif solusi

Kode	Solusi
S-1	Penerapan teknologi *smart contract*, ZKProof, dan SSI
S-2	Pembayaran berbasis NFT dan tokenisasi
S-3	Sistem transaksi dengan *identity federation*, *homomorphic encryption*, dan SMPC

III.3.1.1 Penerapan Teknologi *Smart Contract*, ZKProof, dan SSI

Solusi ini menggunakan kombinasi teknologi *smart contract*, ZKProof dan SSI untuk meningkatkan keamanan dan privasi pengguna dalam P2P AP. *Smart contract* digunakan untuk menangani proses transaksi serta penyelesaian sengketa, SSI memungkinkan pengguna memiliki kendali penuh atas data identitas mereka tanpa melibatkan pihak ketiga, dan ZKProof memungkinkan verifikasi data tanpa mengungkapkan detail sensitif. Selain itu, solusi ini juga mengimplementasikan *booking credential* berbasis SSI dan ZKProof. Pengguna akan menerima *verifiable credential* setelah melakukan pemesanan. *Credential* ini dapat diverifikasi oleh penyedia layanan menggunakan ZKProof tanpa mengungkapkan identitas pengguna, sehingga proses *check-in* tetap sah dan aman tanpa mengorbankan anonimitas. Dengan menggunakan protokol berbasis *blockchain*, solusi ini menawarkan transparansi yang tinggi sekaligus menjaga privasi pengguna.

III.3.1.2 Pembayaran Berbasis NFT dan Tokenisasi

Solusi ini menggunakan NFT (*non-fungible tokens*) dan tokenisasi untuk mengelola transaksi dan pemesanan dalam P2P AP. Pada sistem ini, pengguna akan menerima NFT yang berfungsi sebagai bukti pemesanan atau hak akses ke layanan tertentu, yang tidak dapat dipalsukan. NFT ini bertindak sebagai bukti kepemilikan digital, yang dapat digunakan oleh pengguna untuk mengakses layanan tanpa perlu verifikasi identitas secara langsung. Melalui teknologi *smart contract*, transaksi dan penyelesaian sengketa akan dikelola secara otomatis, memastikan bahwa proses tersebut tetap transparan, aman, dan terdesentralisasi, tanpa melibatkan pihak ketiga. NFT ini tidak hanya berfungsi sebagai bukti transaksi yang sah, tetapi juga memungkinkan verifikasi yang aman tanpa mengungkapkan informasi pribadi pengguna, menjaga privasi dan anonimitas dalam platform.

III.3.1.3 Sistem Transaksi dengan *Identity Federation*, *Homomorphic Encryption*, dan SMPC

Solusi ini menggunakan kombinasi *federated identity*, *homomorphic encryption* (HE), dan *secure multi-party computation* (SMPC) untuk menjamin privasi dan kontrol data pengguna. Sistem *federated identity* memungkinkan pengguna untuk melakukan autentikasi melalui penyedia identitas yang terpercaya (*identity providers*) melalui mekanisme *single sign-on*. Sementara itu, *homomorphic encryption* memastikan bahwa data yang sensitif, seperti detail transaksi dan identitas pengguna, tetap terenkripsi selama seluruh proses transaksi, bahkan saat data tersebut diproses oleh pihak yang terlibat. Hal ini menjaga privasi pengguna tanpa mengurangi kemampuan platform untuk memverifikasi dan memproses transaksi. *Secure multi-party computation* (SMPC) memungkinkan beberapa pihak untuk melakukan perhitungan bersama tanpa mengungkapkan data pribadi setiap pihak, misalnya untuk menghitung total pembayaran atau memverifikasi status transaksi. Semua pihak terlibat dalam perhitungan ini tanpa mengetahui data sensitif satu sama lain. Dengan kombinasi ketiga teknologi ini, P2P AP dapat menyediakan transaksi yang aman, terverifikasi, dan terdesentralisasi, mengurangi ketergantungan pada pihak ketiga, serta meningkatkan privasi pengguna.

III.3.2 Analisis Penentuan Solusi

Untuk memastikan solusi yang dihasilkan dapat menyelesaikan masalah yang telah diidentifikasi sebelumnya, kriteria desain ditetapkan dengan mempertimbangkan hasil identifikasi masalah dan tujuan solusi. Kriteria ini berfungsi sebagai acuan dalam merancang, mengembangkan, dan mengevaluasi solusi secara sistematis. Tabel III.4 menjelaskan kriteria-kriteria desain tersebut yang akan menjadi dasar dalam menilai keberhasilan artefak yang dikembangkan dalam penelitian ini.

Tabel III.4 Kriteria desain

Kode	Kriteria desain	Deskripsi
K-1	Privasi dan anonimitas pengguna	Sejauh mana solusi melindungi data pribadi dan memungkinkan verifikasi tanpa mengungkap identitas.
K-2	Desentralisasi dan kedaulatan data	Tingkat kendali pengguna atas data dan identitas mereka tanpa keterlibatan pihak ketiga.
K-3	Keamanan transaksi dan sengketa	Kemampuan sistem untuk menjaga integritas transaksi dan menyelesaikan sengketa secara aman dan adil.
K-4	Kemudahan implementasi dan integrasi	Tingkat kemudahan dalam mengintegrasikan solusi ke dalam sistem yang ada, mempertimbangkan kompleksitas teknis dan kesiapan infrastruktur.
K-5	Transparansi dan auditabilitas	Seberapa jelas dan dapat diaudit proses yang terjadi, termasuk transaksi, *booking*, dan penyelesaian sengketa.
K-6	*Latency* verifikasi	Waktu yang dibutuhkan untuk memverifikasi *credential*, *booking*, atau transaksi.
K-7	Biaya tambahan	Biaya tambahan untuk satu transaksi, termasuk pemesanan dan verifikasi.

Tabel III.5 menyajikan rangkuman hasil evaluasi dari setiap alternatif solusi berdasarkan kriteria desain yang telah ditentukan, menggunakan skala penilaian 1 hingga 9. Skor 1 menunjukkan bahwa suatu alternatif solusi memiliki tingkat kesesuaian yang sangat rendah terhadap kriteria desain. Skor 9 mencerminkan bahwa solusi tersebut sangat memenuhi kriteria yang ditetapkan dan memiliki potensi tinggi untuk menyelesaikan permasalahan secara efektif.

Tabel III.5 Perbandingan alternatif solusi

Kode kriteria	S-1	S-2	S-3	*Normalized* S01	*Normalized* S02	*Normalized* S03
K-1	9	7	8	0.3750	0.2917	0.3333
K-2	9	8	5	0.4091	0.3636	0.2273
K-3	8	7	9	0.3333	0.2917	0.3750
K-4	7	8	6	0.3684	0.4211	0.3158
K-5	9	8	6	0.3913	0.3478	0.2609
K-6	8	8	6	0.4444	0.4444	0.3333
K-7	6	7	4	0.3529	0.4118	0.2353

Untuk menentukan solusi terbaik secara objektif, digunakan pendekatan *Multi-Criteria Decision Analysis* (MCDA) dengan *Entropy Weight Method* (EWM). Metode ini digunakan untuk menentukan bobot kriteria secara ilmiah dan obyektif berdasarkan tingkat keragaman informasi (diversifikasi) dari setiap kriteria (Zhu, Tian, dan Yan 2020). Analisis MCDA dilakukan dengan melakukan normalisasi *decision matrix*, seperti yang telah terlihat pada Tabel III.5. Skor dari setiap solusi terhadap suatu kriteria dinormalisasi menggunakan metode rasio terhadap total skor dalam kriteria tersebut, seperti yang terlihat pada Rumus III.1.

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \qquad \text{(III. 1)}$$

Untuk menentukan bobot setiap kriteria, perhitungan entropi dilakukan guna mengukur tingkat ketidakpastian atau penyebaran data dari masing-masing kriteria, dengan menggunakan Rumus III.2. Entropi berfungsi sebagai indikator sejauh mana suatu kriteria memberikan informasi yang berbeda antar alternatif solusi. Semakin rendah nilai entropi suatu kriteria, semakin besar kontribusinya terhadap pengambilan keputusan, karena kriteria tersebut dianggap mengandung informasi yang lebih penting dan relevan.

$$e_j = -\frac{1}{\ln n} \sum_{i=1}^{n} p_{ij} \times \ln p_{ij} \qquad \text{(III. 2)}$$

Setelah mendapatkan nilai entropi untuk setiap kriteria, nilai derajat diversifikasi ditentukan untuk mengukur signifikansi relatif suatu kriteria dalam konteks *multi-criteria decision*. Semakin tinggi nilai derajat diversifikasi, semakin penting kriteria tersebut. Nilai derajat diversifikasi dapat ditentukan dengan menggunakan Rumus III.3.

$$d_j = 1 - e_j \qquad \text{(III. 3)}$$

Setelah nilai derajat diversifikasi diperoleh, langkah berikutnya adalah menghitung bobot relatif untuk setiap kriteria. Bobot ini diperoleh dengan menormalisasi nilai derajat diversifikasi menggunakan Rumus III.4, sehingga total bobot dari seluruh kriteria berjumlah 1. Bobot yang dihasilkan akan digunakan untuk mengevaluasi

dan membandingkan alternatif solusi secara objektif berdasarkan tingkat kepentingan masing-masing kriteria.

$$w_j = \frac{d_j}{\sum_{j=1}^{m} d_j} \quad \text{(III. 4)}$$

Hasil penghitungan entropi, derajat diversifikasi, dan bobot akhir untuk setiap kriteria ditunjukkan pada Tabel III.6. Tabel ini memberikan gambaran komprehensif mengenai seberapa besar kontribusi masing-masing kriteria dalam proses pengambilan keputusan. Informasi ini menjadi dasar penting dalam mengevaluasi dan memilih alternatif solusi yang paling sesuai dengan tujuan penelitian.

Tabel III.6 Entropi, diversifikasi, dan bobot akhir untuk setiap kriteria

Kode kriteria	Entropi	Diversifikasi	Bobot akhir
K-1	0.9952	0.0048	0.0572
K-2	0.9741	0.0259	0.3087
K-3	0.9952	0.0048	0.0572
K-4	0.9977	0.0023	0.0279
K-5	0.9876	0.0124	0.1482
K-6	0.9894	0.0106	0.1263
K-7	0.9770	0.0230	0.2745
Total		0.0838	1.0000

Setelah mendapatkan bobot untuk setiap kriteria, nilai normalisasi dari setiap solusi dikalikan dengan bobot kriteria. Skor akhir setiap solusi kemudian diperoleh dengan menjumlahkan seluruh hasil perkalian antara nilai normalisasi dan bobot kriteria. Solusi dengan skor akhir tertinggi dipilih sebagai solusi yang paling optimal dan layak dikembangkan lebih lanjut.

Berdasarkan evaluasi terhadap setiap alternatif solusi dengan mempertimbangkan kriteria desain yang telah ditetapkan, solusi yang menggabungkan penerapan teknologi teknologi *smart contract*, ZKProof, dan SSI dipilih sebagai solusi yang paling optimal untuk dikembangkan. Solusi ini memperoleh skor akhir tertinggi, menunjukkan kesesuaiannya dengan kebutuhan sistem. Detail hasil penghitungan skor akhir tersebut disajikan pada Tabel III.7.

Tabel III.7 Proses penghitungan skor alternatif solusi berdasarkan bobot

Kode kriteria	Bobot	S-1	S-2	S-3
K-1	0.0572	0.3750	0.2917	0.3333
K-2	0.3087	0.4091	0.3636	0.2273
K-3	0.0572	0.3333	0.2917	0.3750
K-4	0.0279	0.3158	0.4211	0.2632
K-5	0.1482	0.3913	0.3478	0.2609
K-6	0.1263	0.3333	0.4444	0.2222
K-7	0.2745	0.3529	0.4118	0.2353
Total		0.3629	0.3571	0.2800

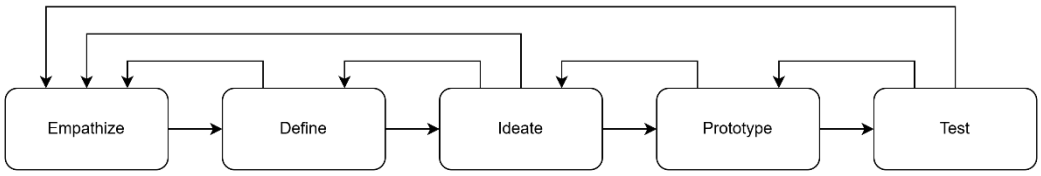
BAB IV

DESAIN SOLUSI

Pada Bab III, telah dibahas tahapan identifikasi masalah dan definisi tujuan yang menjadi dasar dari penelitian ini. Bab ini akan melanjutkan dengan menjelaskan secara lebih terperinci proses desain solusi yang diusulkan. Fokus utamanya mencakup pengembangan prototipe untuk mengatasi masalah dan tantangan yang telah diidentifikasi, serta langkah-langkah pengujian dan validasi yang dilakukan guna memastikan bahwa solusi tersebut efektif, andal, dan sesuai dengan tujuan penelitian.

IV.1 Tahapan Desain

Proses desain dalam penelitian ini akan mencakup pembuatan artefak berupa prototipe P2P AP berbasis *blockchain* yang dirancang untuk mengatasi masalah-masalah yang telah diidentifikasi sebelumnya. Pengembangan solusi dilakukan dengan mengikuti pedoman *Design Science Research Methodology* (DSRM), yang menekankan pentingnya dasar teori yang relevan dan penerapan pengalaman praktis dalam bidang P2P dan *blockchain*. Pada tahap *design and development* pada DSRM, metodologi *design thinking* digunakan untuk mendesain solusi. Hal ini memungkinkan pendekatan yang lebih *user-centered*, yaitu desain difokuskan untuk memenuhi kebutuhan dan tantangan pengguna akhir (Brown dan Funk 2008).



Gambar IV.1 *Design thinking flowchart*

IV.1.1 *Empathize*

Pada tahap *empathize*, fokus utama adalah memahami kebutuhan, perasaan, dan tantangan pengguna secara mendalam. Untuk itu, dilakukan eksplorasi terhadap ekosistem pengguna P2P AP, baik dari sisi *guest*, *host* maupun pengelola platform. Pendekatan ini mencakup studi literatur, pengamatan terhadap platform sejenis, dan analisis kebutuhan pengguna terkait aspek keamanan, privasi, dan kendali atas data pribadi. Dari proses ini ditemukan bahwa salah satu kekhawatiran utama pengguna adalah kurangnya kepercayaan terhadap penyedia platform dalam menyimpan dan mengelola data identitas mereka. Pengguna cenderung enggan memberikan informasi pribadi secara langsung karena adanya risiko kebocoran data, penyalahgunaan identitas, atau sentralisasi kontrol data oleh pihak ketiga.

IV.1.2 *Define*

Setelah mengumpulkan berbagai wawasan dari tahap sebelumnya, pada tahap *define* dilakukan perumusan masalah utama yang perlu diselesaikan. Masalah diformulasikan dalam bentuk *problem statement* yang spesifik dan berpusat pada pengguna. *Problem statement* yang dirumuskan adalah, "Pengguna P2P AP (*guest*) membutuhkan cara untuk melakukan pemesanan dan verifikasi tanpa harus menyerahkan data identitas mereka kepada pihak platform, karena mereka tidak mempercayai platform dalam menyimpan dan mengelola data pribadi secara aman dan tidak ingin kehilangan kendali atas informasi sensitif tersebut." Tahap ini menjadi dasar dalam menentukan fokus pengembangan solusi dan menetapkan kriteria desain yang digunakan dalam evaluasi solusi pada tahap selanjutnya.

IV.1.3 *Ideate*

Tahap *ideate* merupakan fase eksploratif, saat berbagai ide solusi dikembangkan berdasarkan *problem statement* yang telah dirumuskan pada tahap *define*. Proses ini dimulai dengan merumuskan pertanyaan dengan "*How Might We" statement*. Pertanyaan utama yang digunakan dalam tahap ini adalah, "Bagaimana kita dapat

memungkinkan pengguna untuk melakukan pemesanan dan verifikasi tanpa harus memberikan kendali penuh atas data pribadi mereka kepada platform?"

Beberapa ide utama yang dihasilkan dari pertanyaan tersebut beserta analisis penentuan solusi terdapat pada Bab III, khususnya pada Subbab III.3. Penerapan teknologi *smart contract*, ZKProof, dan SSI dianggap paling mampu mengatasi masalah utama pengguna, yaitu ketidakpercayaan terhadap platform dalam menyimpan dan mengelola data pribadi, sekaligus menjaga kontrol penuh atas identitas mereka. Dengan solusi ini, proses transaksi dan verifikasi dapat dilakukan dengan aman, tanpa mengorbankan privasi pengguna. Solusi ini kemudian menjadi dasar untuk tahap selanjutnya, yaitu pengembangan prototipe.

IV.1.4 *Prototype*

Setelah tahap *ideate* menghasilkan solusi terpilih, langkah selanjutnya adalah mengembangkan prototipe. Pada tahap ini, solusi yang dipilih diubah menjadi representasi nyata yang memungkinkan tim untuk menguji dan memvalidasi ide yang telah dikembangkan. Prototipe ini bukanlah versi final dari sistem, melainkan sebuah alat untuk eksplorasi lebih lanjut dan untuk mengidentifikasi potensi masalah dalam desain awal. Prototipe awal dirancang untuk mencakup beberapa elemen inti:

a. fitur membuat reservasi untuk *guest*,
b. integrasi SSI dan ZKProof menggunakan Privado ID, dan
c. *smart contract* untuk otomatisasi transaksi dan penyelesaian sengketa.

Prototipe ini dirancang dengan fokus pada fungsionalitas dasar yang dapat menguji alur pengguna, kehandalan teknologi yang digunakan, dan respons sistem terhadap penggunaan yang nyata. Pengujian ini penting untuk menilai sejauh mana solusi mampu memenuhi kebutuhan pengguna dalam konteks P2P AP berbasis *blockchain*. Prototipe awal dibuat dalam bentuk aplikasi web yang dapat digunakan oleh pengguna untuk melakukan pemesanan dan verifikasi identitas dengan menggunakan SSI dan ZKProof.

IV.1.5 *Test*

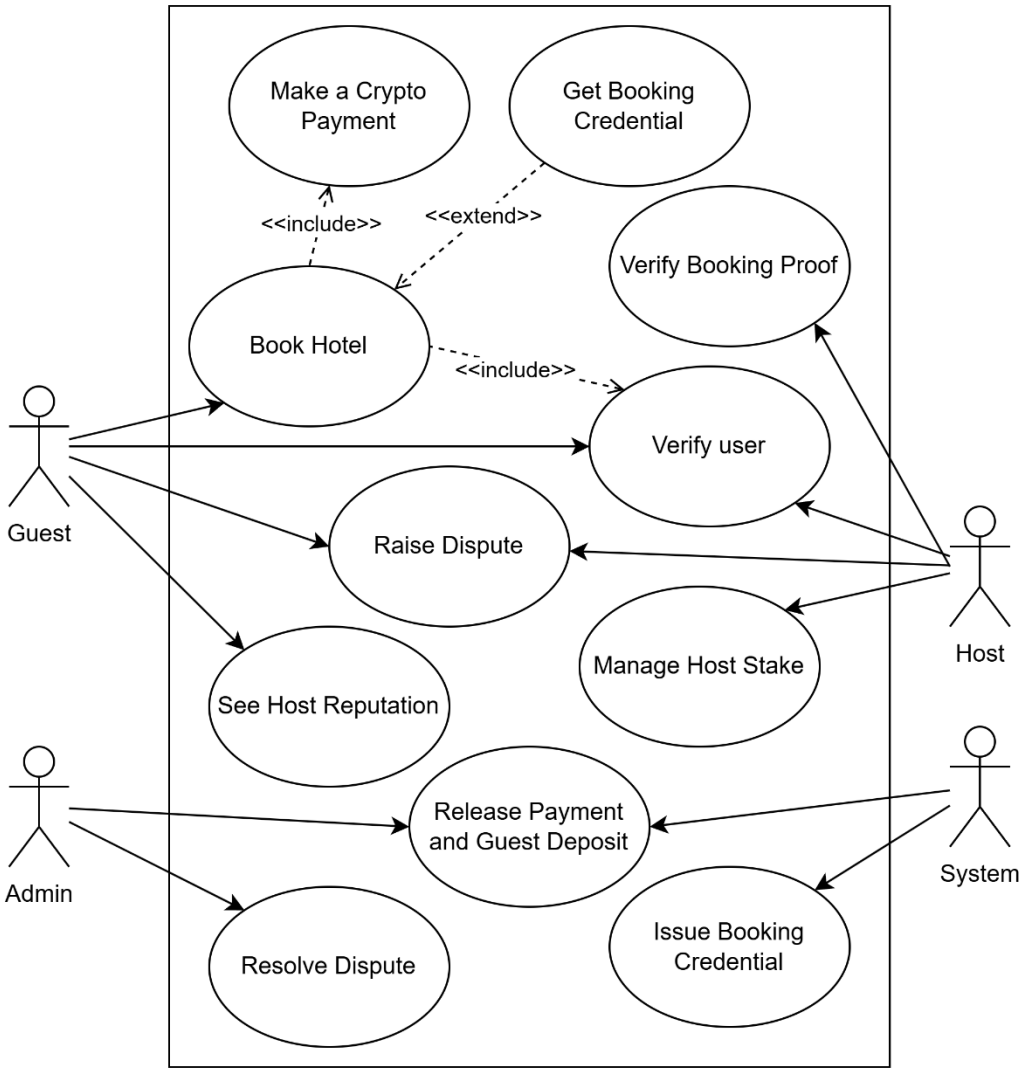
Tahap berikutnya adalah *testing*, yaitu fase saat prototipe yang telah dikembangkan diuji untuk mengevaluasi efektivitas dan kinerja solusi. Pengujian dilakukan dengan melibatkan pengguna yang mewakili audiens target, yang berinteraksi langsung dengan prototipe untuk mengidentifikasi potensi masalah dan mengumpulkan *feedback*. Detail terkait tahap *testing* akan dijelaskan secara lebih komprehensif pada Bab V.

IV.2 Hasil Desain

Subbab ini menyajikan artefak-artefak utama yang dihasilkan dari proses *design thinking*, khususnya pada tahapan *prototype* dan *test*. Artefak-artefak tersebut merupakan representasi konkret dari solusi yang dirancang berdasarkan kebutuhan dan permasalahan pengguna yang telah diidentifikasi sebelumnya. Selain itu, artefak ini juga menjadi acuan penting dalam proses pengembangan sistem secara teknis serta dalam evaluasi awal terhadap efektivitas dan kelayakan solusi yang diusulkan.

IV.2.1 *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan interaksi antara pengguna dan sistem yang diusulkan, serta memperlihatkan cara pengguna dapat mengakses dan menggunakan berbagai fungsi dalam sistem. Diagram ini memberikan visualisasi yang jelas mengenai fungsi-fungsi utama sistem dan peran pengguna. Gambar IV.2 menggambarkan diagram *use case* untuk P2P AP berbasis *blockchain*, yang menunjukkan berbagai *use case* utama yang akan dibuat. Diagram ini membantu mengidentifikasi fungsionalitas yang perlu ada dalam sistem, serta memberikan gambaran yang jelas mengenai proses yang terlibat. Berbagai *use case* yang bersifat generik, seperti autentikasi, pencarian hotel, pembuatan listing untuk *host*, serta fitur pendukung lainnya, tidak dijelaskan dalam diagram ini untuk menjaga kejelasan dan fokus terhadap proses-proses kunci yang membedakan sistem ini dari platform konvensional.



Gambar IV.2 *Use case diagram* P2P AP berbasis *blockchain*

Gambar IV.2 juga menunjukkan keterkaitan antar *use case*, seperti proses verifikasi identitas yang menjadi prasyarat sebelum *guest* dapat melakukan *booking*. Keterkaitan antar *use case* tersebut digambarkan oleh relasi `<<include>>` dan `<<extend>>`. Relasi `<<include>>` menunjukkan bahwa suatu *use case* selalu memanggil atau melibatkan *use case* lainnya sebagai bagian dari prosesnya, sedangkan `<<extend>>` menunjukkan bahwa suatu *use case* dapat memperluas fungsionalitas *use case* lainnya, tetapi hanya dalam kondisi tertentu. Tabel IV.1

menjabarkan use case yang ada pada P2P AP yang akan dibuat, disertai dengan penjelasannya.

Tabel IV.1 Penjelasan *use case*

Use case	Penjelasan
Make a crypto payment	*Guest* melakukan pembayaran menggunakan aset kripto (seperti ETH, dll) untuk memesan hotel. Pembayaran ini dilakukan melalui *smart contract* agar aman dan transparan.
Get booking credential	Setelah melakukan pemesanan hotel, *guest* menerima *verifiable credential* sebagai bukti pemesanan yang dapat disimpan di *wallet* dan digunakan untuk verifikasi saat *check-in*.
Book hotel	*Guest* memilih properti, mengisi detail pemesanan, dan menyelesaikan proses pemesanan.
Verify booking proof	*Host* memverifikasi bahwa guest benar-benar memiliki bukti pemesanan (*booking credential*) tanpa melihat data pribadi tamu secara langsung. Ini dilakukan dengan ZKProof atau melalui *wallet*.
Raise dispute	Jika terjadi masalah (misalnya *host* tidak menyediakan akomodasi sesuai deskripsi), *guest* dapat mengajukan sengketa (*dispute*) melalui sistem.
Verify user	Sistem memverifikasi bahwa pengguna (*guest*/*host*) memiliki *credential* tertentu, seperti telah melakukan KYC, *booking credential*, dll.
See host reputation	*Guest* dapat melihat data reputasi *host* berdasarkan riwayat aktivitas *host* pada platform yang disimpan di *blockchain*.
Manage host stake	*Host* dapat melakukan *deposit* dan *withdraw* sejumlah uang sebagai bentuk jaminan (*stake*) untuk mengurangi risiko penipuan. *Stake* ini akan disimpan di *smart contract*.
Release payment and guest deposit	Setelah masa inap selesai tanpa ada sengketa, *smart contract* secara otomatis melepaskan pembayaran ke *host* dan mengembalikan *deposit* ke *guest*.
Resolve dispute	*Admin* menyelesaikan sengketa yang diajukan oleh *guest* atau *host* dengan memverifikasi bukti-bukti yang ada dan membuat keputusan akhir, termasuk alokasi dana.
Issue booking credential	Sistem menerbitkan *booking credential* berbasis *verifiable credential* kepada *guest* setelah pemesanan berhasil dan pembayaran diterima.

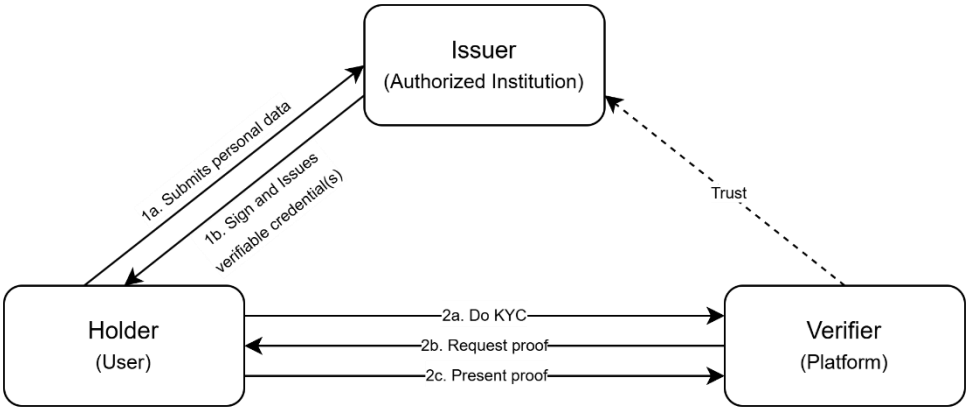
Terdapat empat aktor utama dalam diagram ini, yaitu *Guest*, *Host*, *Admin*, dan *System*. *Guest* adalah aktor yang menggunakan sistem sebagai pemesan, *Host* adalah pihak yang menyediakan properti dan dapat membuat listing baru, *Admin*

adalah aktor yang menangani *dispute* yang terjadi antara *Guest* dan *Host*, dan *System* adalah sistem P2P AP itu sendiri.

IV.2.2 *Communication Diagram*

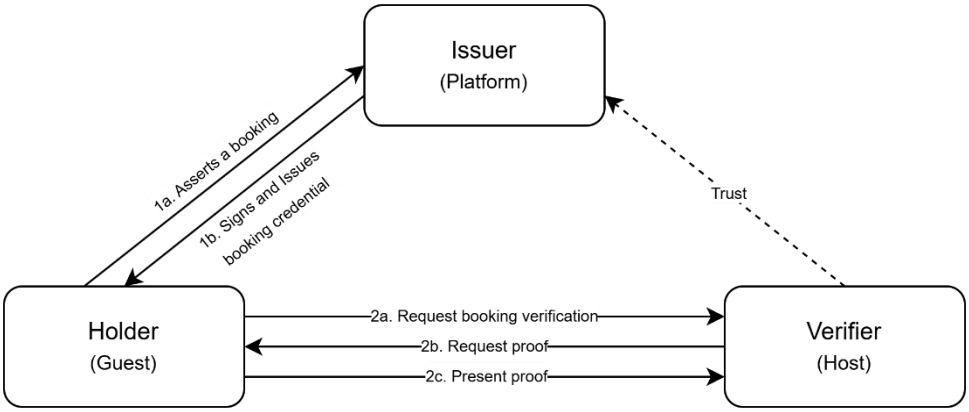
Communication diagram adalah salah satu jenis *interaction diagram* yang menggambarkan cara objek-objek dalam sistem saling berkomunikasi melalui pertukaran pesan, dalam rangka menjalankan suatu proses atau fungsi tertentu. Diagram ini menekankan urutan dan struktur komunikasi antar objek dalam konteks skenario tertentu, serta menunjukkan peran masing-masing objek dalam alur interaksi. Fokus utama dari *communication diagram* adalah memperlihatkan hubungan dan kolaborasi antar aktor atau objek dalam sistem secara visual dan terstruktur.

Secara umum, *communication diagram* yang disajikan pada subbab ini merupakan representasi yang lebih spesifik dan operasional dari model *trust triangle* yang telah disajikan sebelumnya (lihat Gambar II.5). Model *trust triangle* pada Gambar II.5 menggambarkan hubungan konseptual antara tiga aktor utama (*issuer*, *holder*, dan *verifier*) dalam kerangka kepercayaan *identitas digital*, sedangkan *communication diagram* pada Gambar IV.3 dan IV.4 memvisualisasikan bagaimana ketiga entitas tersebut saling berinteraksi secara teknis dalam konteks *use case* tertentu.



Gambar IV.3 *Communication diagram* untuk KYC *credential*

Diagram yang ditunjukkan pada Gambar IV.3 menggambarkan alur komunikasi antara pengguna (*holder*), lembaga yang berwenang (*issuer*), dan platform (*verifier*) dalam proses penerbitan dan pembuktian KYC *credential*. *Holder* mengirimkan data pribadi ke *issuer*, lalu *issuer* menerbitkan *verifiable credential*. Kemudian, *credential* ini digunakan oleh *holder* untuk membuktikan identitasnya kepada *verifier* melalui proses verifikasi berbasis ZKProof.



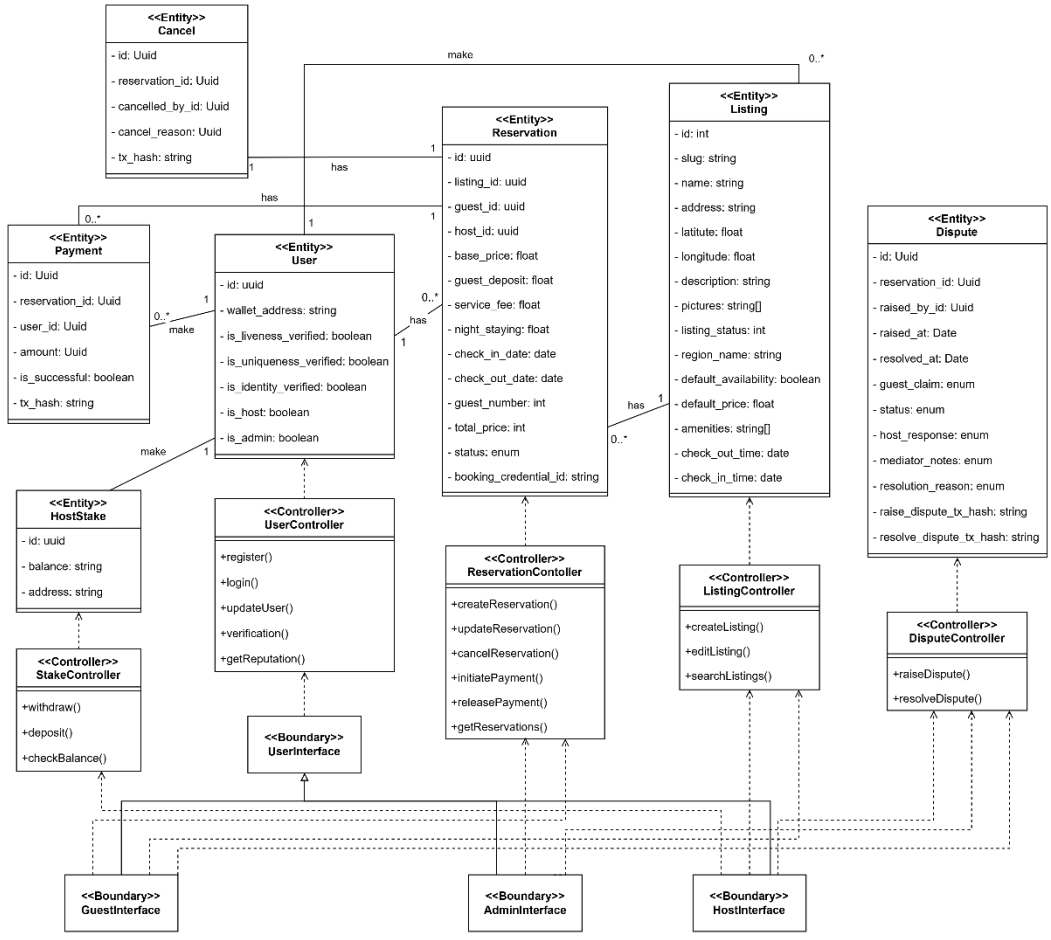
Gambar IV.4 *Communication diagram* untuk *booking credential*

Diagram yang ditunjukkan pada Gambar IV.4 menjelaskan proses komunikasi antara *guest* (*holder*), platform (*issuer*), dan *host* (*verifier*) dalam konteks pembuktian reservasi. *Guest* mengeklaim telah melakukan pemesanan kepada platform, yang kemudian menerbitkan *booking credential*. *Credential* tersebut digunakan oleh *guest* untuk membuktikan reservasi kepada host. Dengan alur seperti ini, data *guest* tidak perlu disimpan dalam *database* untuk membuktikan kepemilikan *guest* atas reservasi yang telah dilakukan.

IV.2.3 *Class Diagram*

Class diagram digunakan untuk merepresentasikan struktur statis dari sistem yang diusulkan, dengan menampilkan kelas-kelas utama, atribut, metode, serta hubungan di antaranya. Diagram ini memberikan gambaran mendalam tentang cara data dan logika bisnis diorganisasikan dalam sistem. Dengan memanfaatkan diagram kelas, pengembang dapat memahami struktur internal sistem, mendefinisikan relasi antarobjek, serta memastikan bahwa desain yang dibuat konsisten dengan

kebutuhan fungsional dan non-fungsional. Gambar IV.5 menggambarkan diagram kelas dari P2P AP berbasis *blockchain* yang mengadopsi model *Entity-Controller-Boundary* (ECB). Model ini dirancang untuk memisahkan tanggung jawab dalam sistem, yang membantu memastikan struktur yang jelas dan memudahkan pengembangan serta pemeliharaan lebih lanjut. Pada diagram ini, setiap komponen memiliki peran spesifik yang berfokus pada aspek tertentu dari sistem, sehingga alur kerja sistem menjadi lebih efisien dan terorganisir.

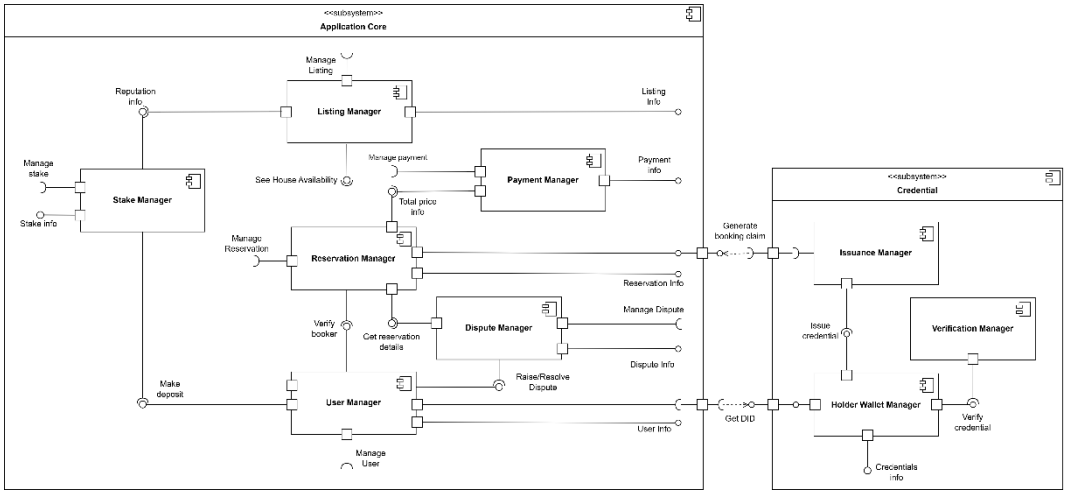


Gambar IV.5 *Class diagram* P2P AP berbasis *blockchain*

IV.2.4 *Component Diagram*

Component diagram digunakan untuk menggambarkan struktur fisik sistem, menampilkan komponen perangkat lunak utama beserta hubungan antar komponen tersebut. Gambar IV.6 menunjukkan pengorganisasian dan interaksi antar komponen dalam sistem untuk mendukung fungsionalitas P2P AP berbasis *blockchain* secara keselurhan. Terdapat dua subsistem pada P2P AP yang dibuat, yaitu *application core* dan *credential*.

Setiap subsistem memiliki komponen masing-masing dan setiap komponen dalam sebuah subsistem memiliki tanggung jawab spesifik dan saling berinteraksi untuk menjalankan proses bisnis. Sebagai contoh, komponen "Reservation Manager" bertanggung jawab mengelola proses reservasi. Dengan adanya diagram komponen, pengembang dapat merencanakan integrasi dan komunikasi antar komponen secara lebih baik, memastikan sistem beroperasi secara efisien dan sesuai kebutuhan.

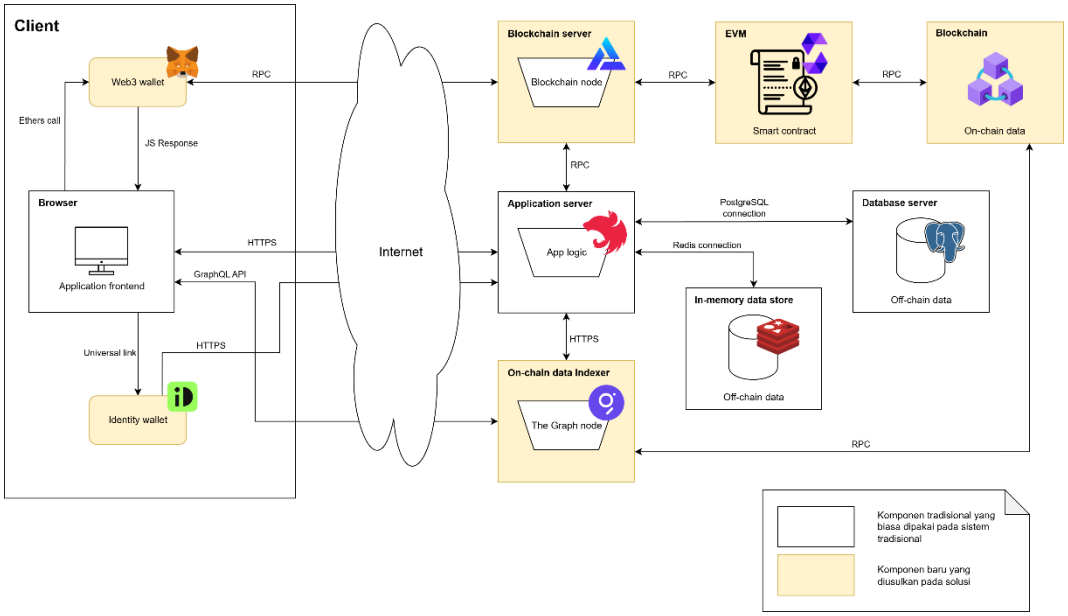


Gambar IV.6 *Component diagram* P2P AP berbasis *blockchain*

IV.2.5 *Architecture Diagram*

Architecture diagram menggambarkan arsitektur sistem, yaitu elemen-elemen dalam sistem dan interaksi antar elemen-elemen tersebut untuk mencapai tujuan sistem secara keseluruhan. Diagram ini memberikan pemahaman visual mengenai

alur komunikasi antar sistem, perangkat keras, serta perangkat lunak yang terlibat. Pada platform berbasis *blockchain*, umumnya terdapat beberapa perbedaan signifikan terkait arsitektur dibandingkan dengan arsitektur tradisional (lihat Gambar III.1). Hal ini disebabkan oleh integrasi teknologi *blockchain*, yang menggantikan beberapa komponen pusat yang ada pada model tradisional dengan desentralisasi data dan transaksi. Meskipun demikian, pada sistem P2P AP berbasis *blockchain* yang akan dibuat masih menggunakan komponen-komponen tradisional. Gambar IV.7 memberikan gambaran secara umum tentang arsitektur sistem P2P AP berbasis *blockchain*.



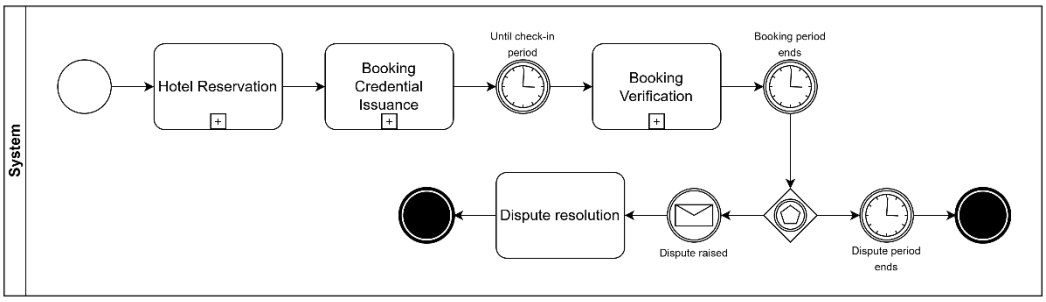
Gambar IV.7 Arsitektur sistem P2P AP berbasis *blockchain*

Komponen-komponen tradisional pada P2P AP berbasis *blockchain* mencakup *application server*, *in-memory data store*, dan *database server*. Komponen-komponen ini berfungsi untuk menangani logika bisnis dan penyimpanan data yang tidak melibatkan *blockchain* secara langsung. Komponen-komponen baru yang diusulkan (berbasis *blockchain*) lebih berfokus pada pengelolaan transaksi, verifikasi *credential*, dan penyimpanan data yang bersifat desentralisasi, sementara komponen tradisional tetap berperan dalam menjalankan proses yang lebih umum dan pengelolaan fungsionalitas *non-blockchain*. Dengan kombinasi kedua

pendekatan ini, sistem dapat memanfaatkan *blockchain* untuk aspek-aspek tertentu sambil mempertahankan keandalan dan efisiensi komponen-komponen tradisional dalam mendukung pengalaman pengguna dan pengelolaan data.

IV.2.6 *Business Process Model and Notation*

BPMN digunakan untuk memodelkan alur proses bisnis secara terperinci, dengan menekankan urutan aktivitas, pengambilan keputusan, dan interaksi antar entitas seperti pengguna dan sistem. Notasi ini dirancang untuk mudah dipahami oleh berbagai pemangku kepentingan, baik dari sisi teknis maupun non-teknis. BPMN memberikan gambaran yang lebih operasional dibandingkan *use case diagram*, sehingga cocok untuk mendeskripsikan cara suatu proses dijalankan dalam sistem.

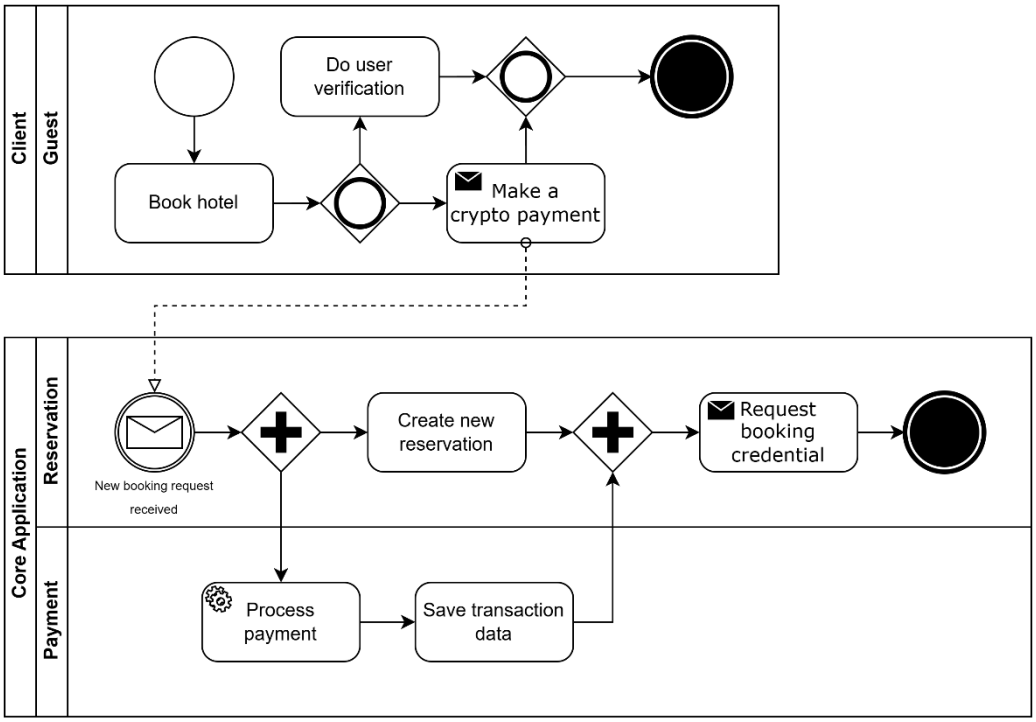


Gambar IV.8 BPMN level 1 sistem P2P AP berbasis *blockchain*

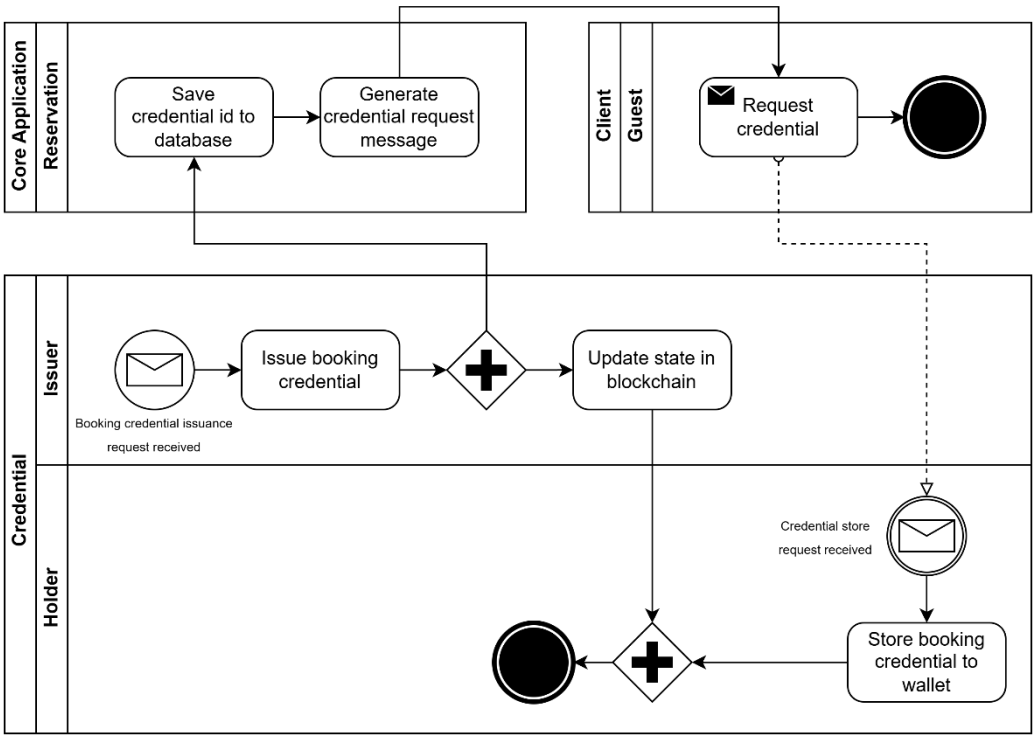
Model BPMN yang dibuat berfokus pada alur *end-to-end* proses *booking* hingga tahap verifikasi reservasi oleh *host*, dengan menampilkan tahapan utama yang dilalui oleh pengguna dan sistem secara menyeluruh. Model ini dirancang untuk memperlihatkan keputusan dan interaksi yang terjadi dalam proses bisnis tersebut. Gambar IV.8 memberikan gambaran umum mengenai proses bisnis pada sistem yang akan dikembangkan, dan disajikan dalam bentuk BPMN level 1 yang menyoroti alur utama tanpa terlalu banyak detail teknis.

BPMN level 2 digunakan untuk menjabarkan proses bisnis secara lebih terperinci dan teknis, sehingga dapat diterjemahkan langsung ke dalam implementasi sistem. Model ini menggambarkan setiap aktivitas, keputusan, dan interaksi dengan tingkat detail yang lebih tinggi dibandingkan BPMN level 1. Gambar IV.9 menunjukkan

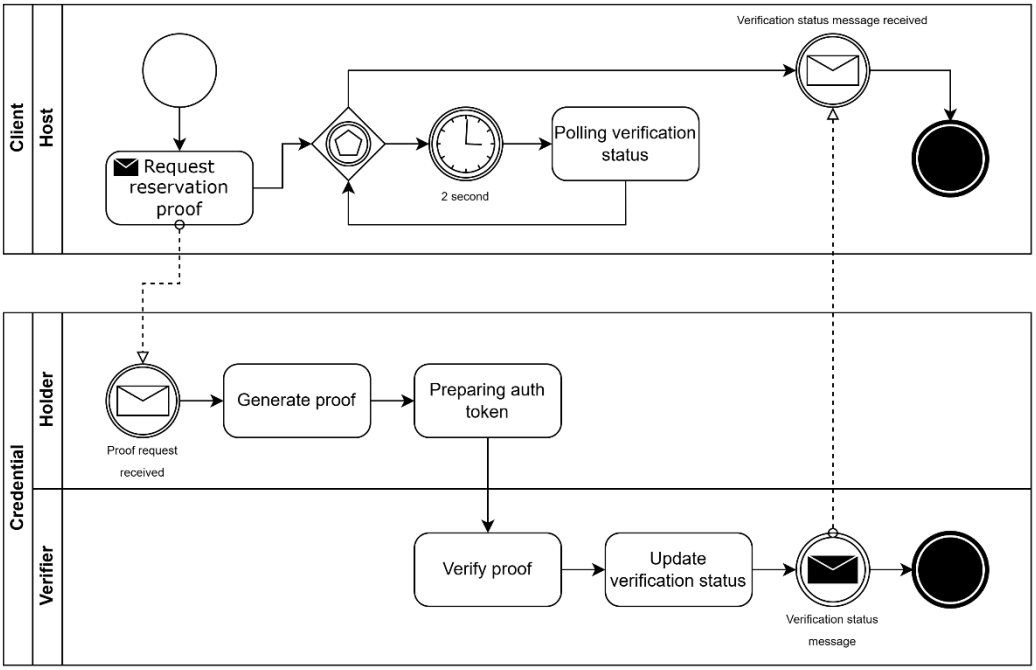
BPMN untuk proses *hotel reservation* oleh pengguna, Gambar IV.10 menggambarkan proses *booking credential issuance*, sedangkan Gambar IV.11 menampilkan proses verifikasi reservasi oleh host sebagai langkah akhir sebelum akomodasi dapat digunakan. Ketiga gambar tersebut menyajikan pemodelan terperinci dari masing-masing proses, yang disusun secara sistematis untuk memastikan kejelasan dalam pengembangan dan integrasi sistem.



Gambar IV.9 BPMN untuk proses *hotel reservation*



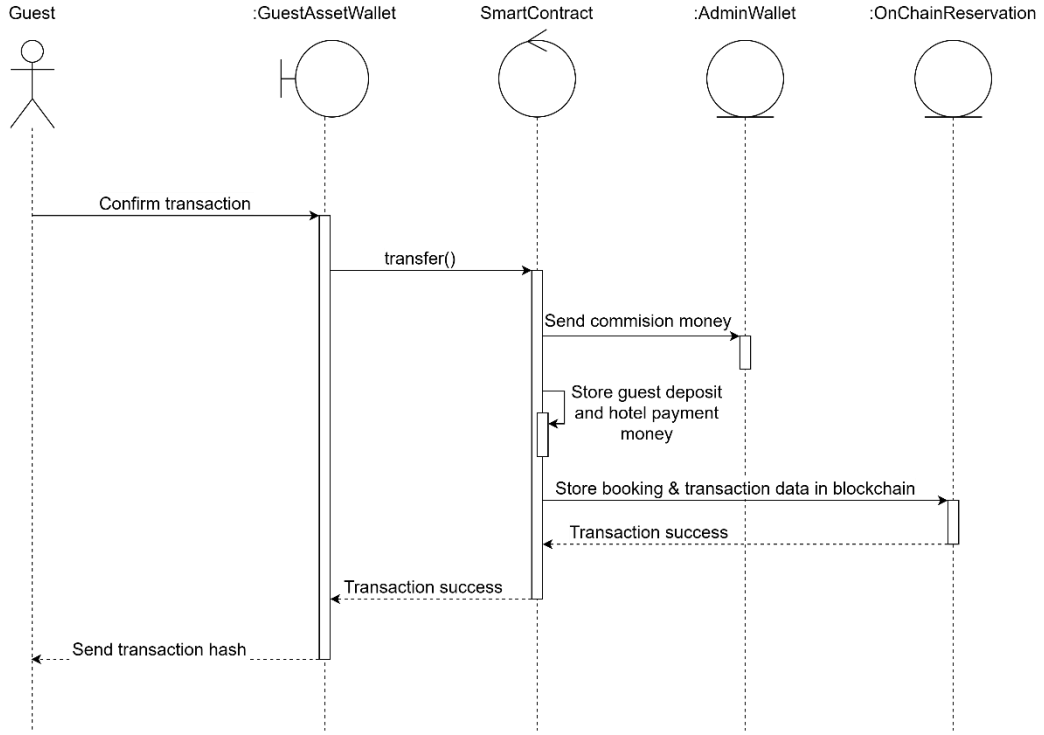
Gambar IV.10 BPMN untuk proses *booking credential issuance*



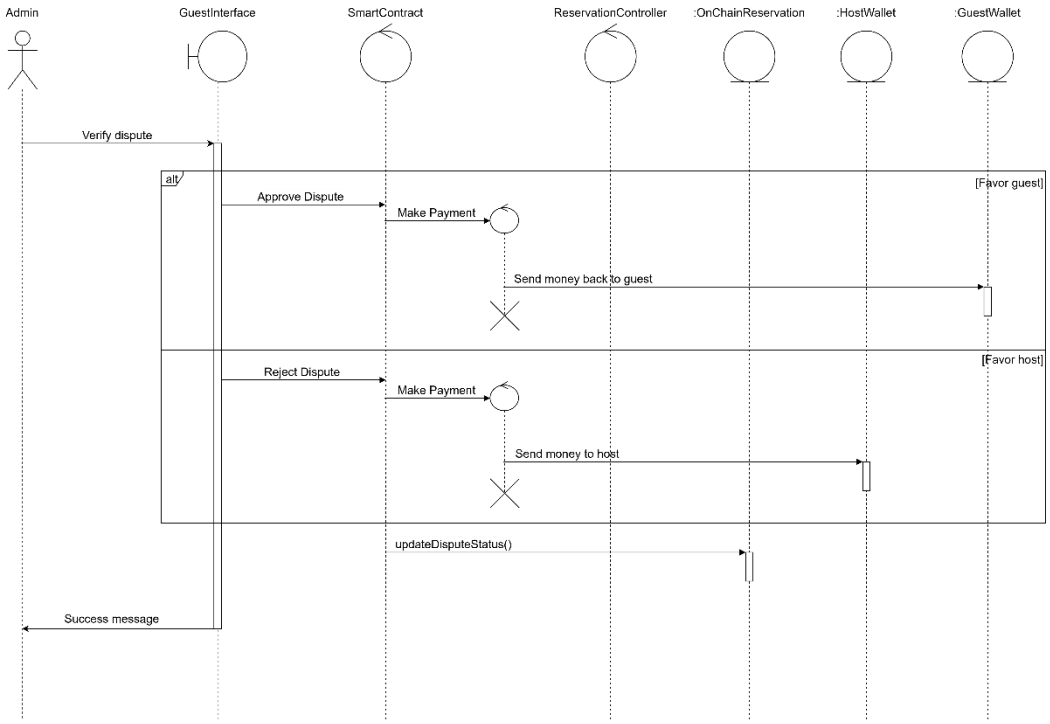
Gambar IV.11 BPMN untuk proses booking credential verification

IV.2.7 *Sequence Diagram*

Sequence diagram merupakan diagram yang digunakan untuk memodelkan interaksi antar objek dalam sistem berdasarkan urutan waktu secara kronologis. Diagram ini menampilkan cara objek atau komponen berkomunikasi satu sama lain dengan mengirimkan pesan secara berurutan untuk menjalankan suatu fungsi atau proses. Dengan menampilkan alur komunikasi dari atas ke bawah sesuai dengan waktu, *sequence diagram* membantu dalam memahami logika eksekusi sistem secara detail dan berurutan.



Gambar IV.12 *Sequence diagram* untuk *make a crypto payment*



Gambar IV.13 *Sequence diagram* untuk *resolve dispute*

Sequence diagram lebih dekat dengan implementasi kode karena menunjukkan cara modul atau *service* berinteraksi secara terperinci, sehingga dapat dipakai sebagai *blueprint* saat proses implementasi. Diagram ini membantu pengembang memahami alur komunikasi antar komponen secara teknis, termasuk urutan pemanggilan fungsi dan pertukaran data. *Sequence diagram* yang dibuat dalam penelitian ini difokuskan secara khusus untuk menggambarkan *use case* yang dianggap kompleks, seperti *make a crypto payment* yang ditunjukkan pada Gambar IV.12 dan *resolve dispute* yang ditunjukkan pada Gambar IV.13.

IV.3 Hasil Implementasi

Subbab ini menyajikan hasil akhir dari proses implementasi sistem P2P AP yang telah dikembangkan berdasarkan kebutuhan fungsional dan rancangan sistem yang telah dijelaskan pada bab sebelumnya. Implementasi dilakukan dengan mengacu pada desain teknis yang telah dirancang untuk memastikan kesesuaian antara spesifikasi dan hasil akhir. Hasil implementasi ini mencakup elemen-elemen teknis yang menjadi komponen utama sistem, seperti integrasi *smart contract*, mekanisme

verifikasi *credential* menggunakan SSI dan ZKProof, serta proses transaksi berbasis *blockchain*.

IV.3.1 Deskripsi Sistem

Inovasi sistem P2P AP yang dikembangkan dalam proyek ini dapat dikategorikan sebagai suatu bentuk *innovation of meaning*, sebagaimana dikemukakan oleh Roberto Verganti dalam karyanya *Overcrowded*. Inovasi ini tidak hanya berfokus pada penciptaan solusi teknis baru (*innovation of solutions*), melainkan pada penciptaan makna baru bagi pengguna dalam berinteraksi dengan sistem pemesanan akomodasi. Tabel IV.2 menunjukkan perbandingan P2P AP tradisional dengan inovasi yang dikembangkan.

Tabel IV.2 Perbandingan P2P AP tradisional dengan inovasi

Aspek	P2P AP tradisional	P2P AP inovasi
Makna kepercayaan	Dibangun melalui otoritas terpusat dan proses KYC manual.	Dibangun melalui pembuktian kriptografis (ZKProof) tanpa otoritas terpusat.
Kontrol atas identitas	Data identitas dikelola dan disimpan oleh platform.	Pengguna memiliki kontrol penuh atas identitas melalui SSI.
Privasi pengguna	Rentan terhadap penyalahgunaan dan kebocoran data.	Terjaga melalui *selective disclosure* dan bukti tanpa pengungkapan.
Peran pihak ketiga	Biasanya sangat tergantung pada pihak ketiga terutama dalam penanganan transaksi.	Diminimalisasi melalui *smart contract* yang bersifat otonom dan transparan.
Kepemilikan reservasi	Ditetapkan melalui *database* internal platform.	Diperkuat dengan *verifiable credential* yang dibuktikan melalui ZKProof.
Reputasi pengguna	Dikendalikan dan dimoderasi oleh platform.	Dibangun secara *on-chain* berdasarkan interaksi dan *outcome* transaksi.
Kepemilikan data	Dipegang oleh platform.	Dipegang oleh pengguna (*self-sovereign*).

Sistem ini memperkenalkan makna baru dalam proses pemesanan akomodasi, dengan menggantikan kepercayaan yang sebelumnya dibangun melalui otoritas terpusat dan proses KYC konvensional, menjadi kepercayaan yang dibangun

melalui arsitektur yang bersifat anonim, terdesentralisasi, dan *trust-free*. Pada platform konvensional, pengguna diharuskan menyerahkan data pribadi mereka kepada entitas terpusat yang bertanggung jawab atas verifikasi identitas, penyimpanan data, dan mediasi transaksi. Hal ini tidak hanya menimbulkan risiko kebocoran dan penyalahgunaan data, tetapi juga menempatkan pengguna dalam posisi yang bergantung pada pihak ketiga untuk memperoleh kepercayaan.

Sebaliknya, sistem P2P AP ini mengandalkan kombinasi teknologi SSI dan ZKProof untuk memungkinkan pengguna membuktikan aspek-aspek penting, seperti keunikan identitas, status verifikasi KYC, dan kepemilikan atas reservasi (*booking credential*), tanpa harus mengungkapkan informasi pribadi mereka. Melalui pendekatan ini, pengguna dapat menunjukkan bukti yang valid atas klaim tertentu secara kriptografis, tanpa menyerahkan data mentah yang mendasarinya. *Booking credential*, dalam hal ini, merupakan *verifiable credential* yang dikeluarkan kepada pengguna dan dapat dibuktikan secara *on-chain* melalui ZKProof saat proses *check-in*, tanpa mengorbankan privasi.

Lebih lanjut, sistem ini menggunakan *smart contract* sebagai komponen utama dalam mengelola transaksi dan interaksi antar pengguna secara otomatis, transparan, dan tidak dapat dimanipulasi. Dengan *smart contract*, sistem dapat menghilangkan peran perantara dalam proses pembayaran dan penyelesaian sengketa, serta mencatat riwayat perilaku pengguna sebagai dasar reputasi, tanpa perlu platform pusat sebagai otoritas tunggal. Reputasi pengguna dibangun berdasarkan interaksi dan *outcome* transaksi sebelumnya yang terekam secara permanen di *blockchain*, sehingga menciptakan sistem insentif dan akuntabilitas yang terbuka.

Dengan demikian, inovasi yang dihadirkan bukan sekadar pada aspek teknis, melainkan pada perubahan makna kepercayaan itu sendiri, dari yang sebelumnya harus menyerahkan kendali kepada pihak ketiga demi kepercayaan, menjadi mampu mengontrol identitas dan reputasinya sendiri dalam lingkungan yang terbuka, aman, dan bebas dari otoritas pusat. Inilah yang menjadikan sistem ini

sebagai bentuk *innovation of meaning*, sebuah redefinisi terhadap nilai fundamental yang ditawarkan kepada pengguna, yaitu privasi, otonomi, dan keamanan identitas digital dalam ekosistem pemesanan akomodasi.

IV.3.2 Lingkungan Implementasi

Lingkungan implementasi menjelaskan perangkat keras dan perangkat lunak yang digunakan selama proses pengembangan dan pengujian sistem P2P AP. Pemilihan lingkungan ini didasarkan pada kebutuhan teknis proyek serta kompatibilitas dengan komponen-komponen sistem seperti *smart contract*, basis data, dan antarmuka pengguna. Tabel IV.3 menyajikan daftar komponen lingkungan implementasi yang digunakan dalam pengembangan sistem.

Tabel IV.3 Lingkungan implementasi

Jenis komponen	*Software*/*hardware* yang digunakan
Development tools	a. Visual Studio Code b. Subgraph Studio c. TablePlus d. Subgraph Studio
Sistem operasi	Windows
DBMS	PostgreSQL
Bahasa pemrograman	a. TypeScript b. Solidity
Query language	a. GraphQL b. SQL
Version control system	GitHub

IV.3.3 *Project Structure*

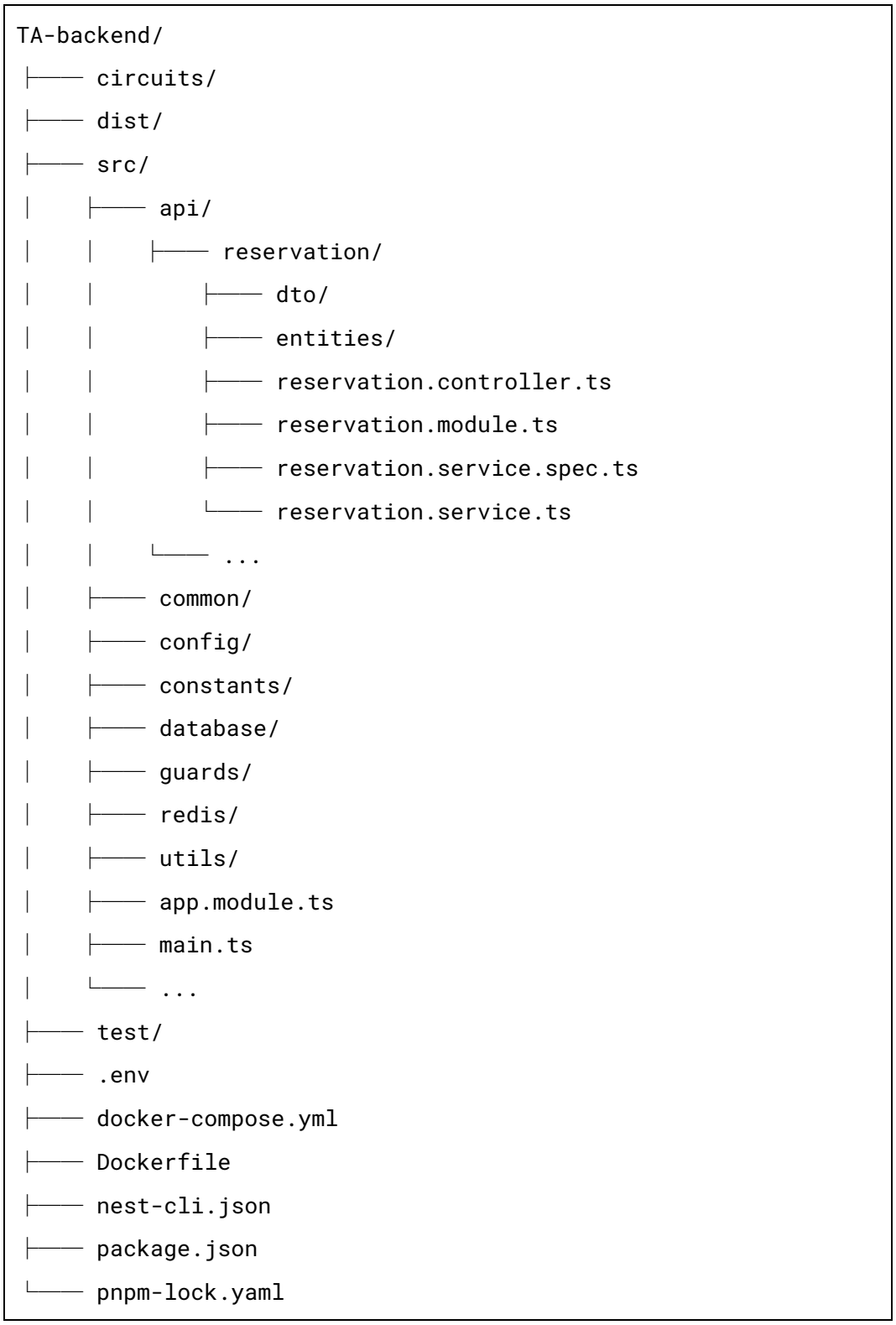
Subbab ini menggambarkan struktur direktori dari empat komponen utama sistem P2P AP berbasis *blockchain* yang dikembangkan, yaitu *frontend*, *backend*, *smart contract*, dan *indexer*. Setiap komponen memiliki peran spesifik dalam mendukung fungsionalitas sistem secara keseluruhan, mulai dari antarmuka pengguna hingga logika bisnis, eksekusi *smart contract*, dan pengindeksan data *blockchain*. Untuk mendukung pengelolaan yang terstruktur dan terpisah antar komponen, proyek ini

dikembangkan dengan pendekatan *polyrepo*, sehingga masing-masing komponen memiliki repositori tersendiri.

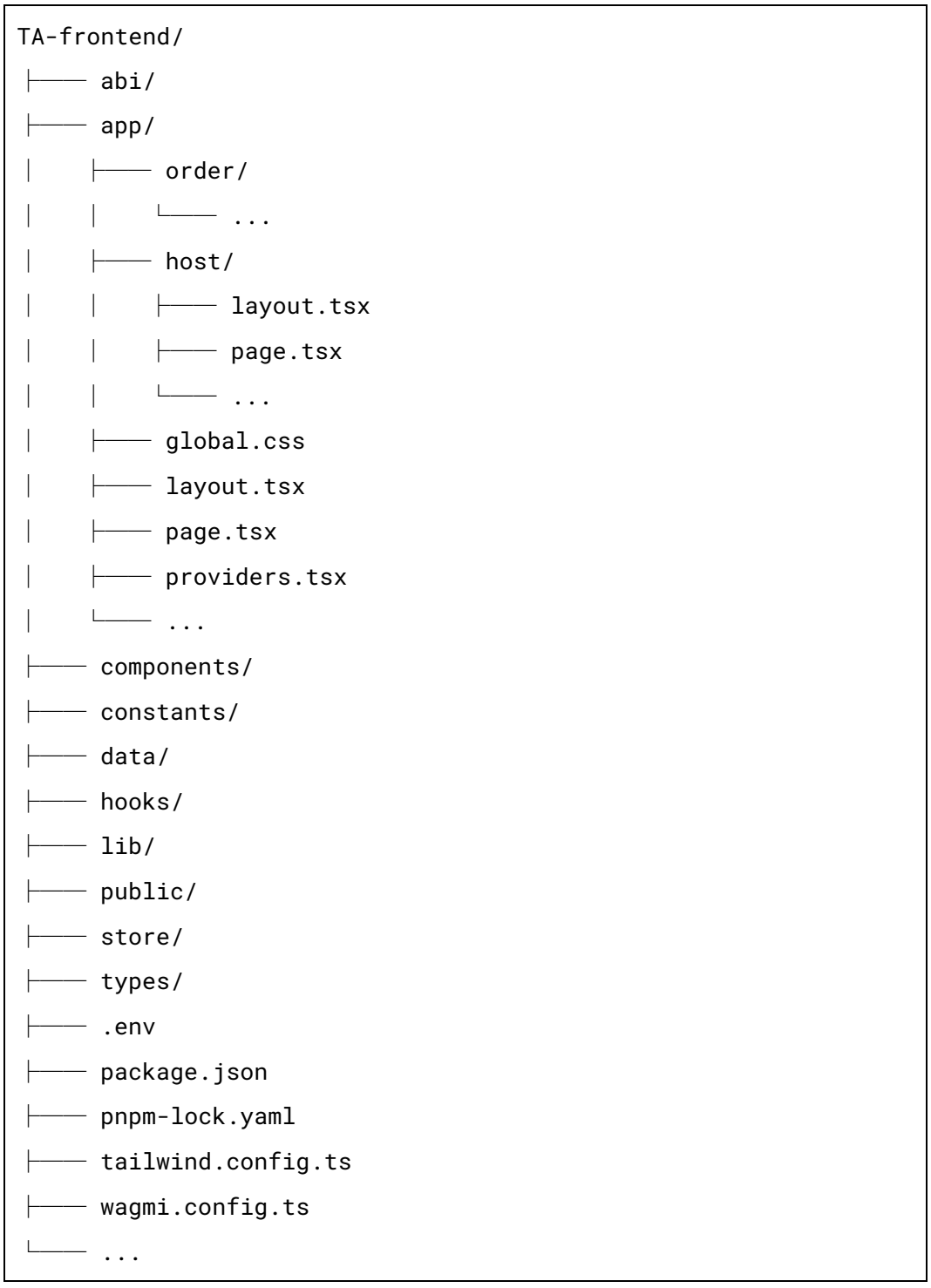
```
TA-smart-contract/
├─── artifacts/
├─── cache/
├─── contracts/
│    └─── HostStake.sol
│    └─── RentalPayments.sol
├─── ignition /
├─── scripts/
│    └─── deploy.js
├─── test/
│    └─── HostStake.js
│    └─── RentalPayments.js
├─── .env
├─── hardhat.config.js
├─── package-lock.json
└─── package.json
```

Gambar IV.14 Strukur direktori komponen *smart contract*

Gambar IV.14 menunjukkan struktur direktori dari repositori pengembangan *smart contract*, yang dikembangkan dengan menggunakan Solidity dan Hardhat. Folder `contracts` berisi kumpulan *smart contract* inti yang digunakan oleh sistem P2P AP, termasuk logika transaksi, pengelolaan reservasi, dan mekanisme *host stake*. Folder `scripts` berisi *script* otomatisasi yang digunakan untuk proses *deployment smart contract* ke jaringan *blockchain*. Selain itu, folder `test` digunakan untuk menampung skenario pengujian terhadap *smart contract*. *File* `hardhat.config.js` merupakan konfigurasi utama proyek yang mendefinisikan versi *compiler* Solidity, jaringan pengujian, serta *plugin* tambahan yang digunakan dalam proses pengembangan.



Gambar IV.15 Struktur direktori komponen *backend*



Gambar IV.16 Struktur direktori komponen *frontend*

Gambar IV.15 menunjukkan struktur direktori dari repositori pengembangan *backend*, yang dikembangkan dengan menggunakan NestJS. Seluruh komponen

utama terletak di folder `src`, termasuk modul-modul API di folder `api`, serta komponen yang dapat di-*reuse* seperti yang ada di folder `src/common`, `src/guards`, dan `src/utils`. Selain itu, terdapat folder `src/api/config`, `src/api/database`, dan `src/api/redis` yang menangani konfigurasi aplikasi, basis data, dan sistem *cache*. Untuk setiap modul API, struktur umumnya terdiri atas tiga komponen utama, yaitu *controller*, *service*, dan *module*, serta dilengkapi dengan folder `dto` untuk mendefinisikan *data transfer object* dan `entities` untuk mendeskripsikan struktur entitas yang digunakan dalam sistem. Selain itu, terdapat folder `circuits` yang berisi sirkuit-sirkuit ZKProof yang akan digunakan untuk proses verifikasi *credential*. Di *project root*, terdapat `Dockerfile` dan `docker-compose.yml` yang berisi konfigurasi *containerization* aplikasi *backend*.

Gambar IV.16 menunjukkan struktur direktori dari repositori pengembangan *frontend*, yang dikembangkan dengan menggunakan Next.js berbasis *app router*. Seluruh logika halaman ditulis di dalam folder `app`. *File* `app/layout.tsx` digunakan untuk mendefinisikan struktur *layout* bersama, sedangkan `app/providers.tsx` digunakan untuk menyusun *global context provider* seperti *react query* dan *wagmi*. Folder `components` berisi komponen UI yang dapat digunakan ulang di berbagai halaman. Folder `abi` menyimpan antarmuka *smart contract* (*application binary interface*) yang digunakan untuk berinteraksi dengan *smart contract*. *File* konfigurasi seperti `tailwind.config.ts` dan `wagmi.config.ts` digunakan untuk mengelola *styling* dengan Tailwind CSS dan integrasi dengan Web3 melalui Wagmi.

Gambar IV.17 menunjukkan struktur direktori dari repositori pengembangan *indexer*, yang dikembangkan dengan menggunakan The Graph. *File* utama `subgraph.yaml` berisi konfigurasi *data source*, *event handler*, dan lokasi ABI *smart contract* yang digunakan. Skema entitas yang diindeks didefinisikan dalam `schema.graphql`. Folder `src` berisi *file* untuk *mapping* seperti `src/rental-payments.ts`, yang berisi logika pemrosesan *event* dari *smart contract*. Folder `generated` menyimpan *file* hasil *generate* otomatis dari Graph CLI, termasuk

`generated/schema.ts` dan *binding* untuk *smart contract* (`generated/RentalPayments`). Folder `abis` berisi *file* ABI *smart contract* yang diperlukan untuk membaca *event*.

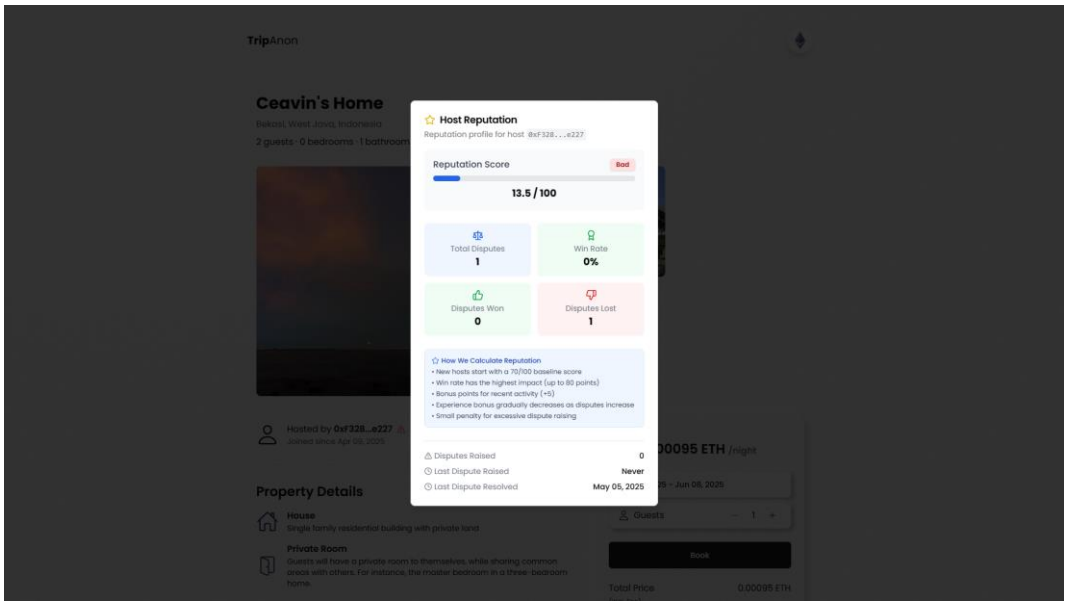
```
TA-subgraph/
├── abis/
├── build/
├── generated/
│   ├── RentalPayments/
│   └── schema.ts
├── src/
│   └── rental-payments.ts
├── docker-compose.yml
├── networks.json
├── package.json
├── schema.graphql
├── subgraph.yaml
└── yarn.lock
```

Gambar IV.17 Struktur direktori komponen *indexer*

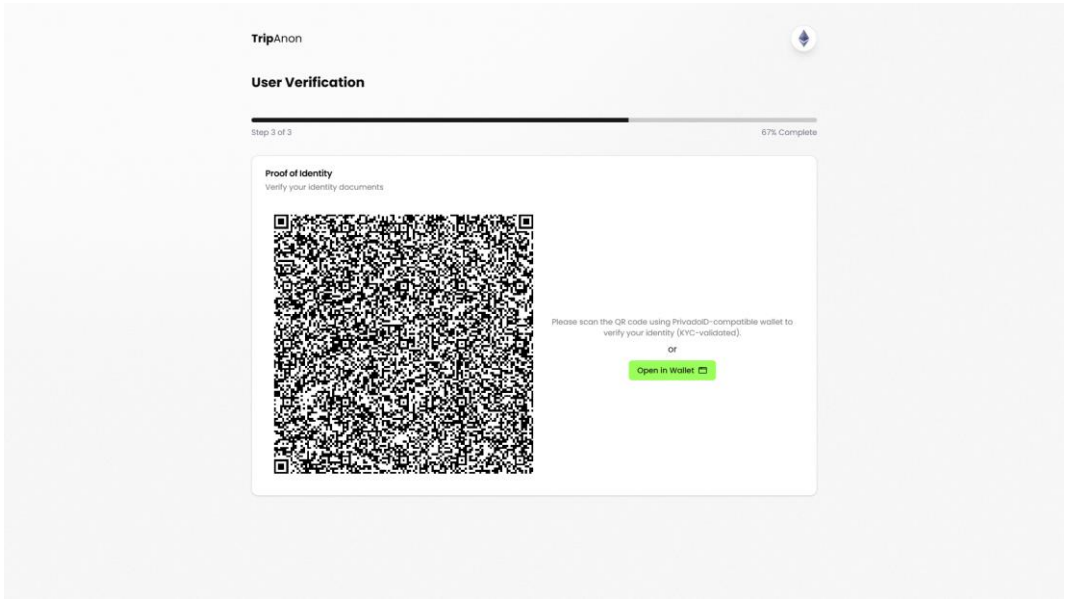
IV.3.4 *User Interface* (UI)

User interface pada sistem P2P AP dirancang untuk memudahkan interaksi pengguna dengan fitur-fitur utama sistem, seperti proses pemesanan, pembayaran, verifikasi identitas, dan verifikasi pemesanan. Meskipun penelitian ini tidak berfokus pada *user interface*, hal ini tetap dibutuhkan untuk menguji fungsionalitas dari sistem yang telah dibuat.

Gambar IV.18 memperlihatkan *pop-up* pada halaman detail hotel, yang menunjukkan reputasi *host* kepada *guest*. Skor reputasi *host* yang ditampilkan dihitung berdasarkan metrik-metrik kuantitatif, seperti *dispute raised*, *dispute won*, dll. Rumus penghitungan skor reputasi *host* juga ditampilkan secara transparan kepada *guest* pada *pop-up* tersebut.



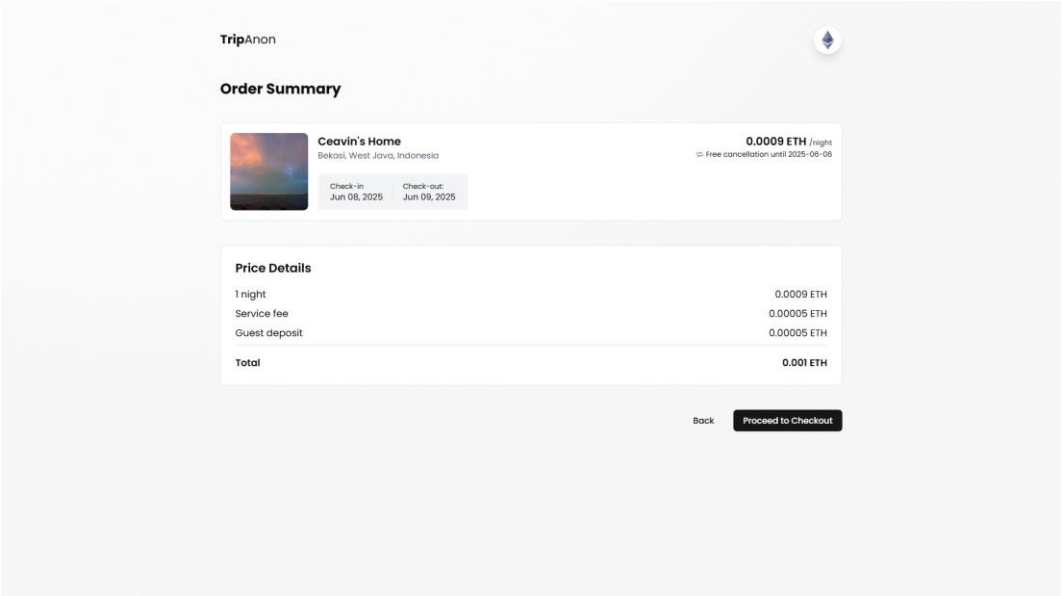
Gambar IV.18 Antarmuka untuk melihat reputasi *host*



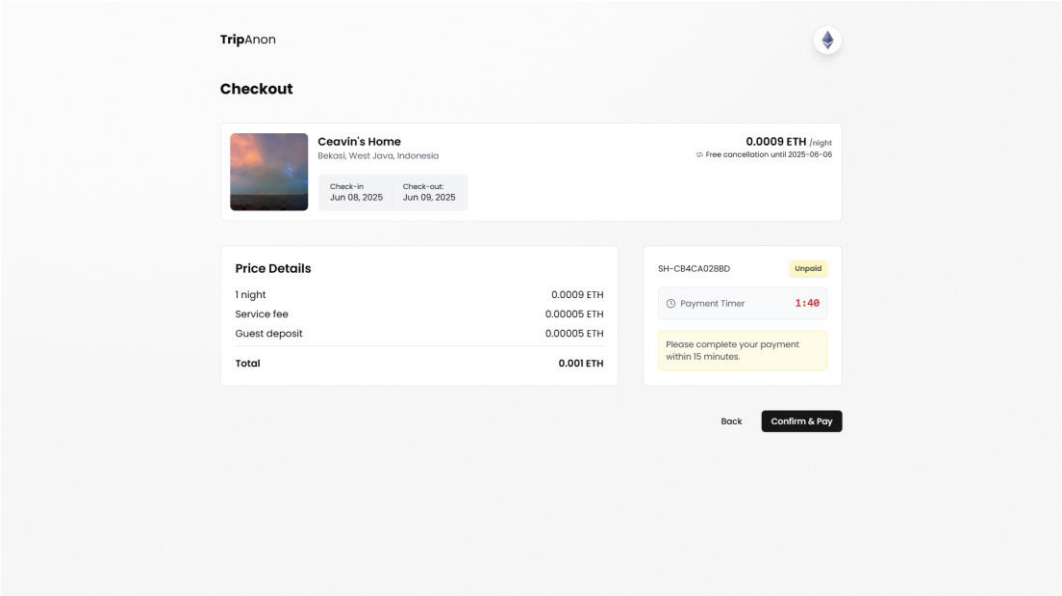
Gambar IV.19 Antarmuka untuk verifikasi pengguna

Gambar IV.19 memperlihatkan halaman verifikasi pengguna, baik *guest* maupun *host*. Pengguna akan melalui tiga tahap verifikasi, yaitu *proof of uniqueness*, *proof of liveness*, dan *proof of identity*. Ketiga tahap ini dirancang untuk memastikan bahwa identitas pengguna valid, unik, dan berasal dari individu yang benar-benar hidup. Pengguna hanya akan dinyatakan terverifikasi apabila telah berhasil melewati seluruh tahapan tersebut secara lengkap.

Gambar IV.20 dan IV.21 masing-masing memperlihatkan halaman detail reservasi dan *checkout*. Pada dasarnya, informasi yang diperlihatkan pada kedua halaman hampir sama. Namun, pada halaman *checkout*, terdapat informasi tambahan seperti *booking number* serta *countdown timer* untuk *guest* menyelesaikan pembayaran sebelum pemesanan kedaluwarsa.

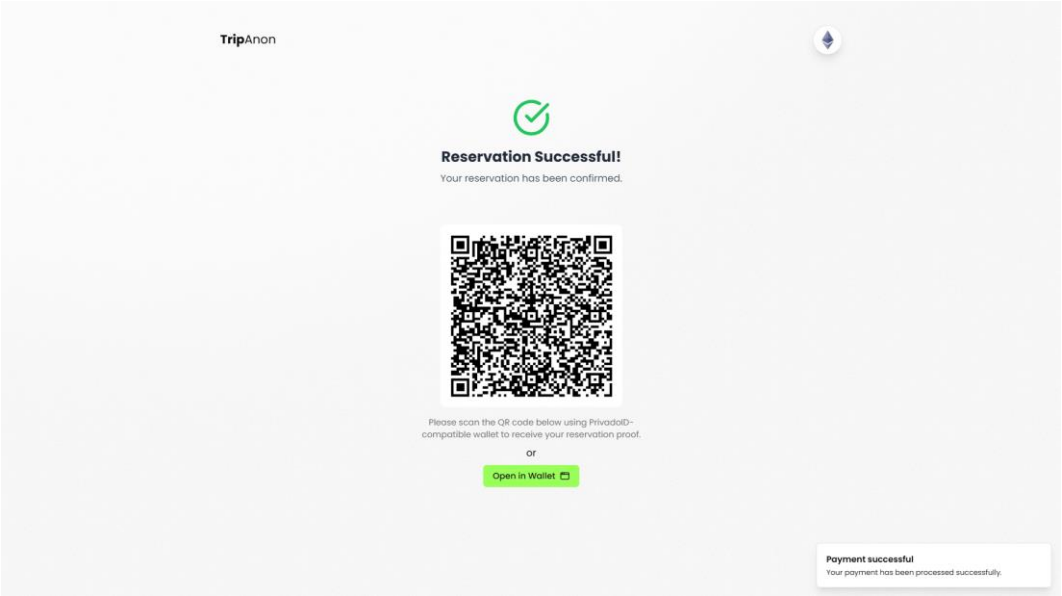


Gambar IV.20 Antarmuka halaman detail reservasi

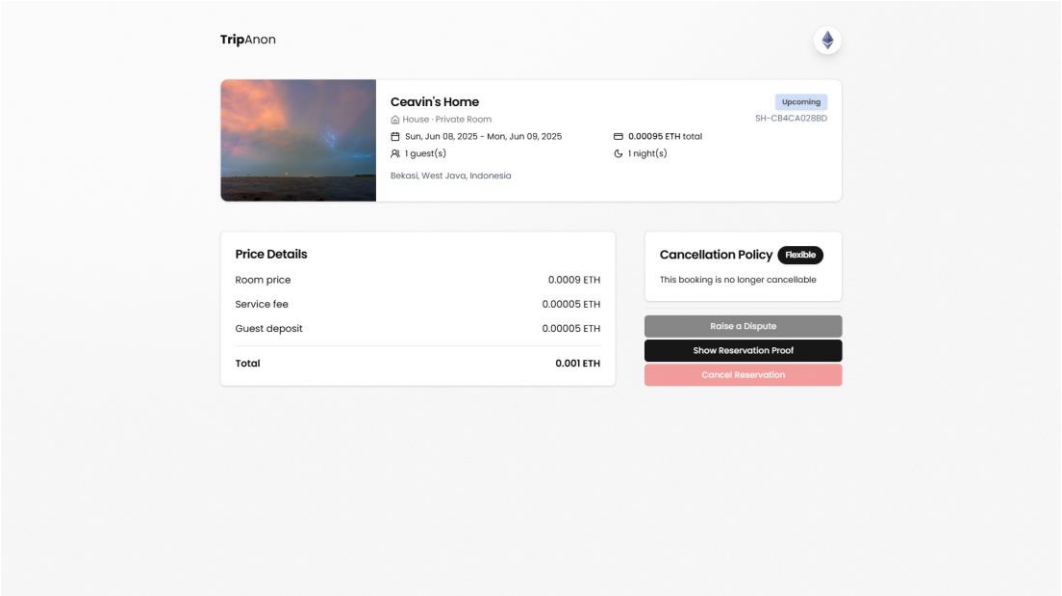


Gambar IV.21 Antarmuka halaman *checkout*

Gambar IV.22 memperlihatkan halaman konfirmasi setelah pemesanan dan pembayaran berhasil. Halaman ini berfungsi sebagai bukti bahwa transaksi telah selesai dan reservasi telah tercatat dalam sistem. Pada halaman ini juga ditampilkan kode QR yang dapat dipindai untuk menyimpan *booking credential* pada *identity wallet* pengguna.



Gambar IV.22 Antarmuka halaman konfirmasi pemesanan berhasil

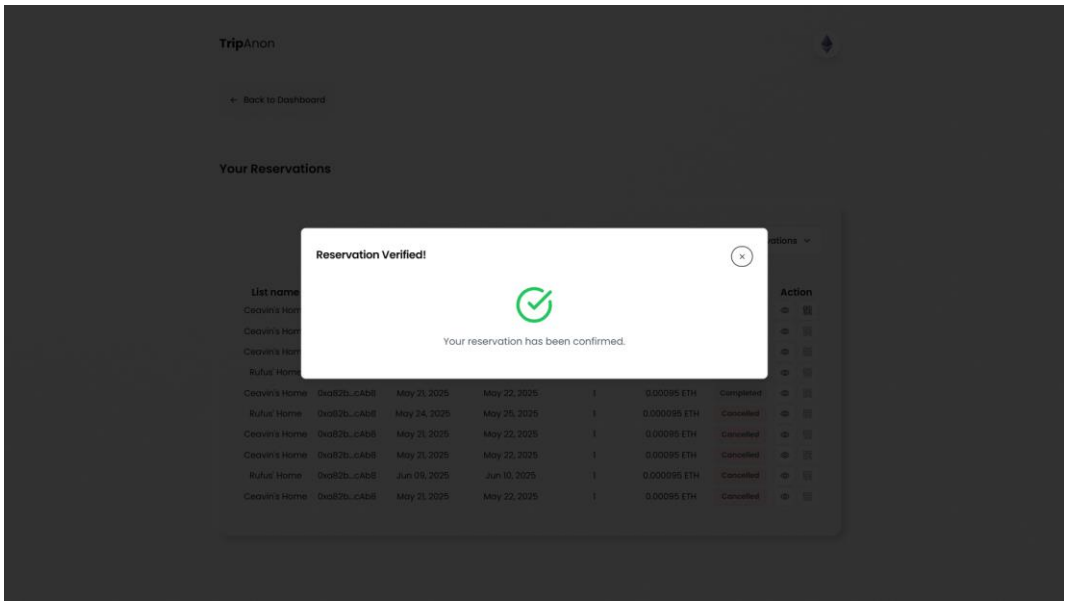


Gambar IV.23 Antarmuka halaman *booking details* untuk *guest*

Gambar IV.23 memperlihatkan halaman detail *booking* untuk *guest*, yang menampilkan informasi *booking* yang telah dibuat. Informasi yang ditampilkan mencakup data akomodasi, tanggal reservasi, detail harga, dan status *booking*. Selain itu, halaman ini juga menyediakan menu bagi *guest* untuk melakukan *dispute*, melihat *reservation proof*, dan serta melakukan pembatalan *booking* jika diperlukan.

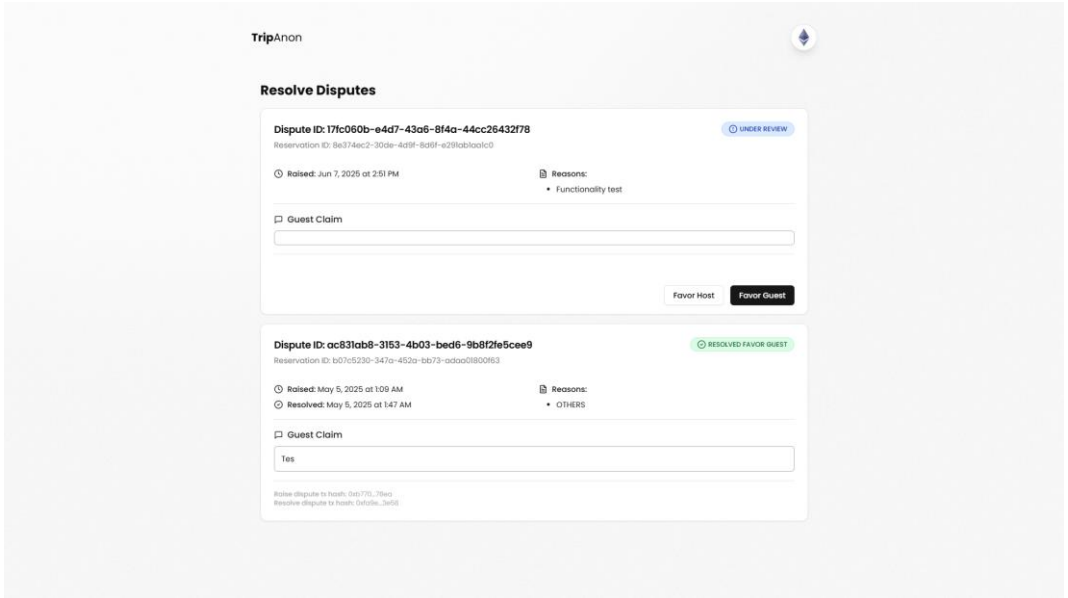
Gambar IV.24 menampilkan halaman detail booking untuk *host*, khususnya saat muncul *pop-up* yang menampilkan kode QR yang dapat dipindai oleh *guest*. Kode QR ini digunakan untuk memverifikasi *booking credential* yang sudah disimpan sebelumnya di *identity wallet* milik *guest*. Proses ini memungkinkan *host* untuk memastikan validitas reservasi tanpa perlu mengakses data pribadi pengguna. Setelah kode QR dipindai dan verifikasi berhasil, *pop-up* akan secara otomatis menampilkan pesan konfirmasi keberhasilan verifikasi, seperti yang ditunjukkan pada Gambar IV.25.

Gambar IV.24 Antarmuka verifikasi *booking*



Gambar IV.25 Antarmuka *booking* terverifikasi

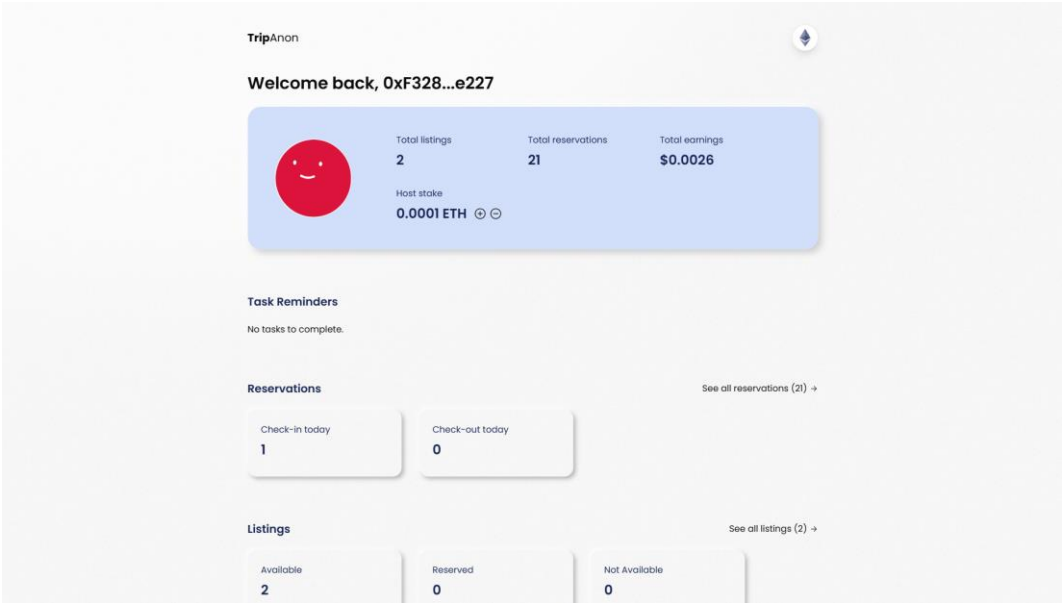
Gambar IV.26 menampilkan halaman *dispute resolution* untuk *admin*. Halaman ini menampilkan daftar semua pesanan yang memiliki sengketa untuk diselesaikan. *Admin* dapat memutuskan sengketa dengan cara berpihak pada *host* atau *guest*, berdasarkan informasi yang ada.



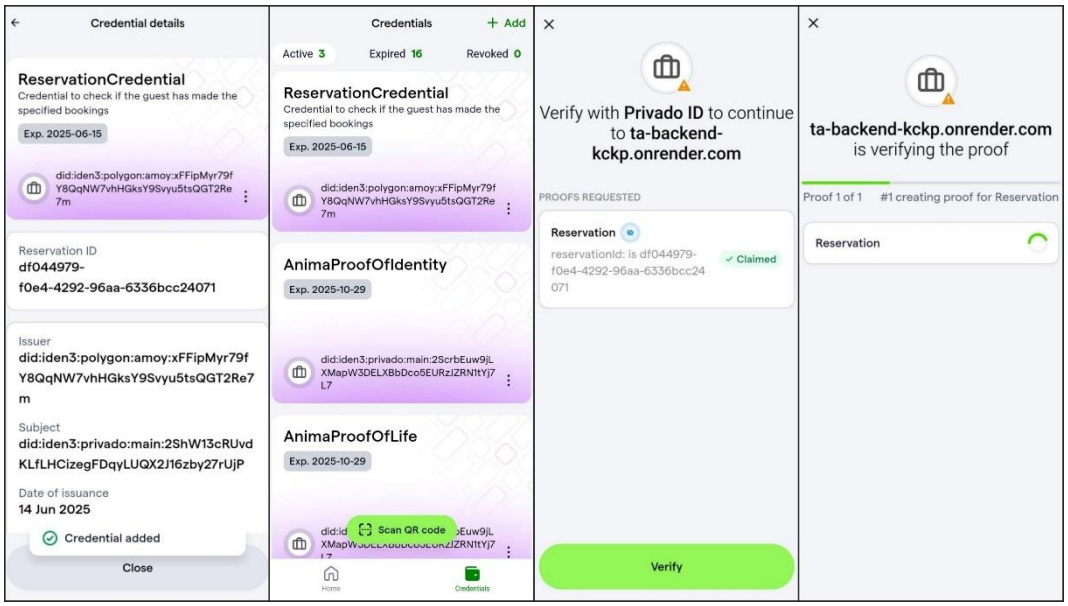
Gambar IV.26 Antarmuka halaman *dispute resolution* untuk *admin*

Gambar IV.27 menampilkan halaman *host dashboard*. Pada halaman ini, ditampilkan ringkasan statistik terkait *hosting*, seperti jumlah token yang sudah di-*deposit*, jumlah reservasi yang diterima, jumlah hotel yang telah di-*listing*, dll. Pada halaman ini *host* dapat menggunakan berbagai fitur seperti *host stake deposit*, *host stake withdrawal*, pengelolaan hotel, pengecekan status reservasi, dll.

Gambar IV.28 menampilkan antarmuka *identity wallet* yang disediakan oleh Privado ID, untuk menyimpan serta memverifikasi *credential*, seperti *booking credential*. *Wallet* ini berperan penting dalam ekosistem identitas terdesentralisasi yang diimplementasikan dalam sistem. Dalam sistem ini, *identity wallet* ini digunakan untuk memindai kode QR yang ditampilkan pada Gambar IV.22 (saat *guest* menyimpan *booking credential*) dan Gambar IV.24 (saat *host* memverifikasi *booking credential* yang telah dibawa oleh *guest*).



Gambar IV.27 Antarmuka halaman *host dashboard*



Gambar IV.28 Antarmuka *identity wallet* Privado ID

IV.3.5 Deployment

Proyek yang telah dikembangkan telah berhasil di-*deploy* dan dapat diakses oleh publik. Antarmuka pengguna (*frontend*) di-*deploy* menggunakan layanan Vercel, sistem *backend* di-*hosting* melalui platform Render, dan *smart contract* di-*deploy* pada jaringan Ethereum Sepolia. Tabel IV.4 menunjukkan tautan yang dapat digunakan untuk mengakses *smart contract* dan *frontend* yang sudah di-*deploy*.

Tabel IV.4 Tautan *deployment*

Komponen	Tautan
RentalPayments *smart contract*	https://eth-sepolia.blockscout.com/address/0x0229E7dB6195A98D02aAe044C40D452e9f559A22
HostStake *smart contract*	https://eth-sepolia.blockscout.com/address/0xFf1a346fE650cef5100240A91c96A063CDe72C84
Frontend	https://trip-anon.vercel.app/

BAB V

EVALUASI

V.1 Desain dan Lingkungan Evaluasi

Subbab ini menjelaskan konfigurasi dan kondisi pengujian yang digunakan untuk memastikan bahwa proses evaluasi sistem berjalan secara terkontrol, terstruktur, dan dapat direproduksi. Informasi yang disampaikan mencakup spesifikasi perangkat keras dan perangkat lunak yang digunakan selama pengujian, termasuk versi sistem operasi, *browser*, dan alat pengujian yang relevan. Selain itu, dijelaskan pula jenis data yang digunakan, lingkungan sistem tempat pengujian dijalankan, serta rencana pengujian yang mencakup skenario, tujuan, dan metrik evaluasi yang diterapkan.

V.1.1 Konfigurasi Sistem

Sebelum melakukan evaluasi terhadap sistem P2P AP berbasis *blockchain*, terlebih dahulu dilakukan perancangan konfigurasi pengujian yang mencakup sistem, perangkat keras dan lunak, data, partisipan, serta lingkungan pengujian yang digunakan. Evaluasi dilakukan untuk memastikan bahwa sistem berjalan sesuai kebutuhan fungsional dan non-fungsional, serta memberikan pengalaman pengguna yang baik. Sistem yang dievaluasi merupakan prototipe P2P AP yang dibangun dengan teknologi berbasis web dan *blockchain*. Sistem terdiri dari beberapa komponen utama, yaitu:

a. *Frontend* dibangun menggunakan Next.js 15 dan di-*deploy* di Vercel;
b. *Backend* dibangun menggunakan NestJS dan di-*hosting* di Render (*dockerized*);
c. *Smart contract* dibuat menggunakan Solidity dan di-*deploy* di jaringan Ethereum Sepolia;
d. SSI diimplementasikan dengan menggunakan Privado ID JavaScript SDK;

e. *Database* menggunakan PostgreSQL dan di-*hosting* menggunakan Supabase;
f. Sistem autentikasi menggunakan MetaMask *wallet*.

V.1.2 Perangkat Keras dan Lunak

Beberapa perangkat keras dan perangkat lunak digunakan oleh penguji dalam proses evaluasi prototipe P2P AP untuk memastikan sistem berjalan sesuai dengan spesifikasi yang dirancang. Penggunaan berbagai komponen ini bertujuan untuk menciptakan kondisi pengujian yang realistis dan representatif terhadap lingkungan pengguna sesungguhnya. Tabel V.1 menjabarkan secara terperinci komponen perangkat keras dan perangkat lunak yang digunakan selama proses evaluasi, termasuk spesifikasi komputer, sistem operasi, *browser*, serta alat bantu pengujian yang relevan.

Tabel V.1 Perangkat keras dan lunak yang digunakan dalam evaluasi

Komponen	Spesifikasi
Laptop	a. CPU: AMD Ryzen 7 b. RAM: 12 GB c. OS: Windows 11 d. Node.js v20.18.1
Smartphone/tablet	a. OS: Android 14 b. RAM: 8 GB (*smartphone*)/12 GB (tablet) c. Storage: 256 GB d. Privado ID *wallet compatible*
Browser	a. Microsoft Edge v136.0.3240.64 b. Sudah ter-*install* MetaMask *extension*
Testing tool	a. Postman v11.44.0 b. Jest c. Chai
Static analysis tool	Slither

V.1.3 Data yang Digunakan

Data yang digunakan dalam proses evaluasi sistem terdiri dari dua jenis, yaitu data *dummy* (fiktif) yang menyerupai kondisi nyata dan data riil. Data *dummy* digunakan untuk mensimulasiksan berbagai skenario penggunaan sistem, seperti proses

pemesanan, *dispute handling*, dan skenario lainnya. Data *dummy* yang digunakan dalam pengujian antara lain

a. listing properti fiktif;
b. transaksi pemesanan, pembayaran, dan *dispute* simulatif;
c. credential SSI berupa *booking credential* dan *user verification*.

Sementara itu, untuk menguji aspek interoperabilitas dan integrasi sistem dengan infrastruktur SSI, sebagian data yang digunakan bersifat riil. Contohnya adalah penggunaan DID milik penguji yang telah terdaftar di Privado ID *wallet*. Hal ini dilakukan agar proses *issuance* dan verifikasi *credential* dapat berlangsung secara aktual dan merepresentasikan kondisi produksi (*production-like testing*).

V.1.4 Pihak yang Telibat

Evaluasi sistem dilakukan oleh pengembang sistem, baik dalam *functional testing*, *unit testing*, maupun *smart contract static analysis*. Pada *functional testing*, penguji berperan sebagai tiga aktor utama dalam sistem, yaitu *guest*, *host*, dan *admin*. Setiap aktor menjalankan serangkaian skenario pengujian untuk memastikan bahwa fitur-fitur utama bekerja sesuai dengan yang diharapkan. Sebagai *guest*, penguji melakukan tindakan seperti melakukan pemesanan, menerima *booking credential*, dan membatalkan reservasi. Sebagai *host*, penguji melakukan tindakan seperti memverifikasi *booking credential*, menerima atau menolak pemesanan, dan melakukan *host stake deposit*. Sebagai *admin*, penguji melakukan tindakan seperti *release payment* dan *resolve dispute*.

V.1.5 Lingkungan Pengujian

Lingkungan pengujian menentukan tempat dan kondisi sistem dijalankan selama evaluasi. Uji coba dilakukan dalam dua jenis lingkungan yang merepresentasikan fase pengembangan dan *staging* sebelum produksi. Sistem diuji dalam dua tahap, yaitu:

a. Pengujian lokal (*development environment*), dilakukan menggunakan *localhost*;

b. Pengujian *production* (*staging environment*), dilakukan setelah sistem di-deploy pada *public environment* dengan HTTPS, sehingga dapat diakses oleh berbagai perangkat dan pengguna eksternal.

Pada kedua jenis lingkungan, *smart contract* di-*deploy* di jaringan *testnet*. Penggunaan *testnet* memungkinkan seluruh proses pengujian dilakukan tanpa risiko finansial karena menggunakan token uji coba yang tidak bernilai nyata. Dengan demikian, transaksi dapat disimulasikan secara aman dan efisien sebelum sistem dipindahkan ke jaringan *mainnet* untuk penggunaan sebenarnya.

V.2 Rencana Evaluasi

Subbab ini menjelaskan rencana evaluasi yang disusun untuk memastikan sistem P2P AP yang telah diimplementasikan dapat memenuhi kebutuhan pengguna dan berjalan sesuai dengan fungsionalitas yang telah ditentukan. Rencana evaluasi disusun berdasarkan *use case* utama dari sistem. Setiap rencana evaluasi yang terdapat pada Tabel V.2 didesain untuk merepresentasikan kondisi dunia nyata yang mungkin terjadi dalam penggunaan platform, baik dari sisi *guest*, *host*, maupun *admin*.

Evaluasi dalam proyek ini dilakukan melalui tahap *testing* untuk mengevaluasi kesesuaian terhadap kebutuhan fungsional, serta dan *auditing* untuk mengevaluasi kesesuaian terhadap kebutuhan non-fungsional. *Testing* dilakukan melalui dua pendekatan utama, yaitu *black box testing* menggunakan metode *functional testing* dan *white box testing* dengan metode *unit testing*. Pemilihan metode pengujian disesuaikan dengan karakteristik setiap skenario. Skenario yang berfokus pada interaksi pengguna dan alur sistem diuji menggunakan *functional testing*, sedangkan skenario yang berfokus pada *logic* di balik layar, seperti *backend* atau *smart contract*, memerlukan *unit testing* untuk memastikan kebenaran logika internal dan aspek keamanan sistem. Pada beberapa kasus, kedua metode diterapkan secara bersamaan guna memperoleh hasil evaluasi yang lebih komprehensif. *Auditing* dilakukan dengan pendekatan *static analysis* pada *smart contract*.

Tabel V.2 Rencana evaluasi sistem

Use case	Tujuan pengujian	Jenis pengujian	Kode
Make a crypto payment	Memastikan sistem dapat memproses pembayaran *cryptocurrency* dengan benar.	*Functional testing*	FT-01
	Memastikan kebenaran logika dalam pemrosesan pembayaran dalam *smart contract*.	*Unit testing*	UT-01
	Mendeteksi kerentanan keamanan, *bug* potensial, dan praktik pemrograman yang tidak efisien secara statis pada *smart contract*.	*Static analysis*	SA-01
Get booking credential	Memastikan pengguna dapat memperoleh *booking credential* yang sah.	*Functional testing*	FT-02
Book hotel	Memastikan proses pemesanan hotel berjalan sesuai prosedur dan validasi.	*Functional testing*	FT-03
	Memastikan kebenaran logika saat pemrosesan reservasi dalam *smart contract*.	*Unit testing*	UT-01
	Memastikan kebenaran logika saat pemrosesan reservasi dalam *backend*.	*Unit testing*	UT-02
	Mendeteksi kerentanan keamanan, *bug* potensial, dan praktik pemrograman yang tidak efisien secara statis pada *smart contract*.	*Static analysis*	SA-01
Verify booking proof	Memastikan sistem dapat memverifikasi bukti *booking* secara sah.	*Functional testing*	FT-04
	Memastikan kebenaran logika saat memverifikasi *booking proof* dalam backend.	*Unit testing*	UT-03
Raise dispute	Memastikan pengguna dapat mengajukan sengketa dengan data yang diperlukan.	*Functional testing*	FT-05
	Memastikan kebenaran logika saat *dispute raising* dalam *smart contract*.	*Unit testing*	UT-01
	Mendeteksi kerentanan keamanan, *bug* potensial, dan praktik pemrograman yang tidak efisien secara statis pada *smart contract*.	*Static analysis*	SA-01

Tabel V.3 Rencana evaluasi sistem (lanjutan)

Use case	Tujuan pengujian	*Jenis pengujian*	Kode
Verify user	Memastikan sistem dapat memverifikasi *credential* pengguna atau *host* dengan akurat.	*Functional testing*	FT-06
See host reputation	Memastikan pengguna dapat melihat data reputasi *host*.	*Functional testing*	FT-07
Manage host stake	Memastikan *host* dapat melakukan *deposit* dan *withdraw* uang sebagai jaminan.	*Functional testing*	FT-08
	Memastikan kebenaran logika saat manajemen *host stake* dalam *smart contract*.	*Unit testing*	UT-04
	Mendeteksi kerentanan keamanan, *bug* potensial, dan praktik pemrograman yang tidak efisien secara statis pada *smart contract*.	*Static analysis*	SA-01
Release payment and guest deposit	Memastikan kebenaran logika saat memberikan uang ke setiap pengguna yang terlibat dalam *smart contract*.	*Unit testing*	UT-01
	Mendeteksi kerentanan keamanan, *bug* potensial, dan praktik pemrograman yang tidak efisien secara statis pada *smart contract*.	*Static analysis*	SA-01
Resolve dispute	Memastikan sengketa dapat diselesaikan melalui alur sistem yang benar.	*Functional testing*	FT-09
	Memastikan kebenaran logika saat *dispute resolution* dalam *smart contract*.	*Unit testing*	UT-01
	Mendeteksi kerentanan keamanan, *bug* potensial, dan praktik pemrograman yang tidak efisien secara statis pada *smart contract*.	*Static analysis*	SA-01
Issue booking credential	Memastikan sistem dapat menerbitkan *booking credential* yang sah.	*Functional testing*	FT-02
	Memastikan kebenaran logika penerbitan *booking credential* dalam *backend*.	*Unit testing*	UT-05

V.3 Hasil Evaluasi

Subbab ini menyajikan hasil evaluasi dari sistem yang telah dikembangkan berdasarkan desain yang telah dirancang sebelumnya. Evaluasi dilakukan untuk menilai sejauh mana sistem memenuhi kebutuhan fungsional dan non-fungsional yang telah ditentukan pada tahap perancangan. Hasil dari setiap pengujian dianalisis untuk memastikan bahwa sistem berjalan sesuai dengan ekspektasi dan dapat menangani alur penggunaan secara *end-to-end*.

V.3.1 Hasil Pengujian FT-01

Tabel V.3 menunjukkan hasil *functional testing* pada FT-01. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa sistem dapat menjalankan fungsi pembayaran menggunakan *cryptocurrency* sesuai dengan ekspektasi.

Tabel V.4 Hasil pengujian FT-01

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
a. Reservasi valid; b. Saldo: 0.0845 ETH.	1. Menekan tombol "Confirm & Pay" pada *website*; 2. Menyetujui transaksi melalui MetaMask *wallet*.	**Amount:** 0.00009 ETH	a. Transaksi berhasil dan hash transaksi tercatat di blockchain; b. Muncul notifikasi sukses.	a. Transaksi berhasil dan hash transaksi tercatat di blockchain; b. Sistem menampilkan pesan sukses.	✓
a. Reservasi valid; b. Saldo: 0 ETH.		**Amount:** 0.00009 ETH	a. Transaksi gagal; b. Muncul notifikasi transaksi gagal.	a. Transaksi gagal; b. Muncul notifikasi transaksi gagal.	✓

V.3.2 Hasil Pengujian FT-02

Tabel V.4 menunjukkan hasil *functional testing* pada FT-02. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem dapat mengeluarkan *valid booking credential* yang dapat diterima pengguna setelah transaksi berhasil.

Tabel V.5 Hasil pengujian FT-02

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
a. Reservasi valid; b. Pembayar-an sukses.	1. Membuka halaman bukti reservasi; 2. Memindai kode QR yang ditampilkan menggunakan Privado ID *wallet*.	DID yang telah ditautkan dengan akun.	a. Bukti reservasi sukses disimpan pada wallet; b. Detail *credential* ditampilkan pada *wallet*.	a. Bukti reservasi sukses disimpan pada wallet; b. Detail *credential* ditampilkan pada *wallet*.	✓
a. Reservasi valid. b. Pembayar-an sukses.		DID berbeda dengan yang telah ditautkan dengan akun	a. Bukti reservasi gagal disimpan pada *wallet*; b. Muncul notifikasi gagal.	a. Bukti reservasi gagal disimpan pada *wallet*; b. Muncul notifikasi gagal.	✓

V.3.3 Hasil Pengujian FT-03

Tabel V.5 menunjukkan hasil *functional testing* pada FT-03. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem dapat mencatat pemesanan hotel secara tepat dan menyimpan informasi reservasi ke dalam sistem dengan baik.

Tabel V.6 Hasil pengujian FT-03

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
Status reservasi: *Waiting for payment*.	1. Membuka halaman *checkout*; 2. Menekan tombol "Confirm & Pay" pada *website*;	-	Melanjutkan ke pembayaran dengan MetaMask *wallet*.	Melanjutkan ke pembayaran dengan MetaMask *wallet*.	✓
Status reservasi: *Cancelled*.		-	a. Gagal melanjutkan ke pembayaran; b. Muncul notifikasi gagal.	a. Gagal melanjutkan ke pembayaran; b. Muncul notifikasi gagal.	✓

V.3.4 Hasil Pengujian FT-04

Tabel V.6 menunjukkan hasil *functional testing* pada FT-04. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem berhasil memverifikasi bukti *booking* secara otomatis dan akurat tanpa membuka data sensitif pengguna.

Tabel V.7 Hasil pengujian FT-04

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
Sudah memiliki *booking credential* yang sesuai.	1. Membuka halaman detail reservasi untuk; 2. Memindai kode QR yang ditampilkan menggunakan Privado ID *wallet*.	-	a. Tampilan *proof generation* pada *wallet*; b. Muncul notifikasi reservasi berhasil dikonfirmasi.	a. Tampilan *proof generation* pada *wallet*; b. Muncul notifikasi reservasi berhasil dikonfirmasi.	✓
Tidak memiliki *booking credential* yang sesuai.		-	Muncul pemberitahuan "*credential does not exist*" pada *wallet*.	Muncul pemberitahuan "*credential does not exist*" pada *wallet*.	✓

V.3.5 Hasil Pengujian FT-05

Tabel V.7 menunjukkan hasil *functional testing* pada FT-05. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem mendukung pengajuan *dispute* dengan mencatat alasan dan informasi tambahan yang dibutuhkan untuk proses penyelesaian sengketa.

Tabel V.8 Hasil pengujian FT-05

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
Dalam rentang *dispute period*.	1. Membuka halaman detail reservasi; 2. Menekan tombol "Raise a Dispute"; 3. Melengkapi *dispute form*.	**Reasons:** ["OTHERS"] **Claim:** "Tes"	a. Tombol "Submit Dispute" dapat ditekan; b. Melanjutkan ke konfirmasi transaksi pada MetaMask *wallet*.	a. Tombol "Submit Dispute" dapat ditekan; b. Melanjutkan ke konfirmasi transaksi pada MetaMask *wallet*.	✓
		-	Tombol "Submit Dispute" tidak dapat ditekan.	Tombol "Submit Dispute" tidak dapat ditekan.	✓
Tidak dalam rentang *dispute period*.	Membuka halaman detail reservasi;	-	Tombol "Raise a Dispute" tidak dapat ditekan;	Tombol "Raise a Dispute" tidak dapat ditekan;	✓

V.3.6 Hasil Pengujian FT-06

Tabel V.8 menunjukkan hasil *functional testing* pada FT-06. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem dapat memverifikasi keunikan, *personhood*, dan identitas pengguna.

Tabel V.9 Hasil pengujian FT-06

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
Sudah memiliki *credential* yang sesuai.	1. Membuka halaman verifikasi user; 2. Memindai kode QR yang ditampilkan menggunakan Privado ID *wallet*.	-	a. Tampilan *proof generation* pada *wallet*; b. Muncul notifikasi *credential* berhasil dikonfirmasi.	a. Tampilan *proof generation* pada *wallet*; b. Msuncul notifikasi *credential* berhasil dikonfirmasi.	✓
Tidak memiliki *credential* yang sesuai.		-	Muncul pemberitahuan "*credential does not exist*" pada *wallet*.	Muncul pemberitahuan "*credential does not exist*" pada *wallet*.	✓

V.3.7 Hasil Pengujian FT-07

Tabel V.9 menunjukkan hasil *functional testing* pada FT-07. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem berhasil menampilkan skor reputasi *host* secara lengkap dan akurat berdasarkan metrik yang telah didefinisikan sebelumnya.

Tabel V.10 Hasil pengujian FT-07

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
-	1. Membuka halaman detail hotel; 2. Menekan tombol "Host Reputation"	-	Muncul *pop-up* yang menampilkan skor reputasi *host*.	Muncul *pop-up* yang menampilkan skor reputasi *host*.	✓

V.3.8 Hasil Pengujian FT-08

Tabel V.10 menunjukkan hasil *functional testing* pada FT-08. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem mampu memungkinkan *host* untuk melakukan melakukan *staking* dan mencatatnya dengan baik dalam sistem.

Tabel V.11 Hasil pengujian FT-08

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
Saldo: 0.2634 ETH.	1. Memasukkan jumlah uang yang ingin di-*deposit*; 2. Menyetujui transaksi melalui MetaMask *wallet*.	**Amount:** 0.0003 ETH	a. Muncul notifikasi sukses; b. Jumlah uang yang di-*deposit* bertambah 0.0003 ETH; c. Saldo berkurang 0.0003 ETH.	a. Sistem menampilkan pesan sukses; b. Jumlah uang yang di-*deposit* bertambah 0.0003 ETH; c. Saldo berkurang 0.0003 ETH.	✓
Saldo: 0.2631 ETH.		**Amount:** 1 ETH	Muncul notifikasi transaksi gagal.	Muncul notifikasi transaksi gagal.	✓
Saldo: 0.2631 ETH.	1. Memasukkan jumlah uang yang ingin di-*deposit*; 2. Menyetujui transaksi melalui MetaMask wallet.	**Amount:** 0.0002 ETH	a. Muncul notifikasi sukses; b. Jumlah uang yang di-*deposit* berkurang 0.0002 ETH; c. Saldo bertambah 0.0002 ETH.	a. Muncul notifikasi sukses; b. Jumlah uang yang di-*deposit* berkurang 0.0002 ETH; c. Saldo bertambah 0.0002 ETH.	✓
Saldo: 0.2633 ETH.		**Amount:** 1 ETH	Muncul notifikasi transaksi gagal.	Muncul notifikasi transaksi gagal.	✓

V.3.9 Hasil Pengujian FT-09

Tabel V.11 menunjukkan hasil *functional testing* pada FT-09. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa sistem mampu memungkinkan *admin* menyelesaikan sengketa secara transparan dan hasil keputusan dapat diproses ke dalam *smart contract*.

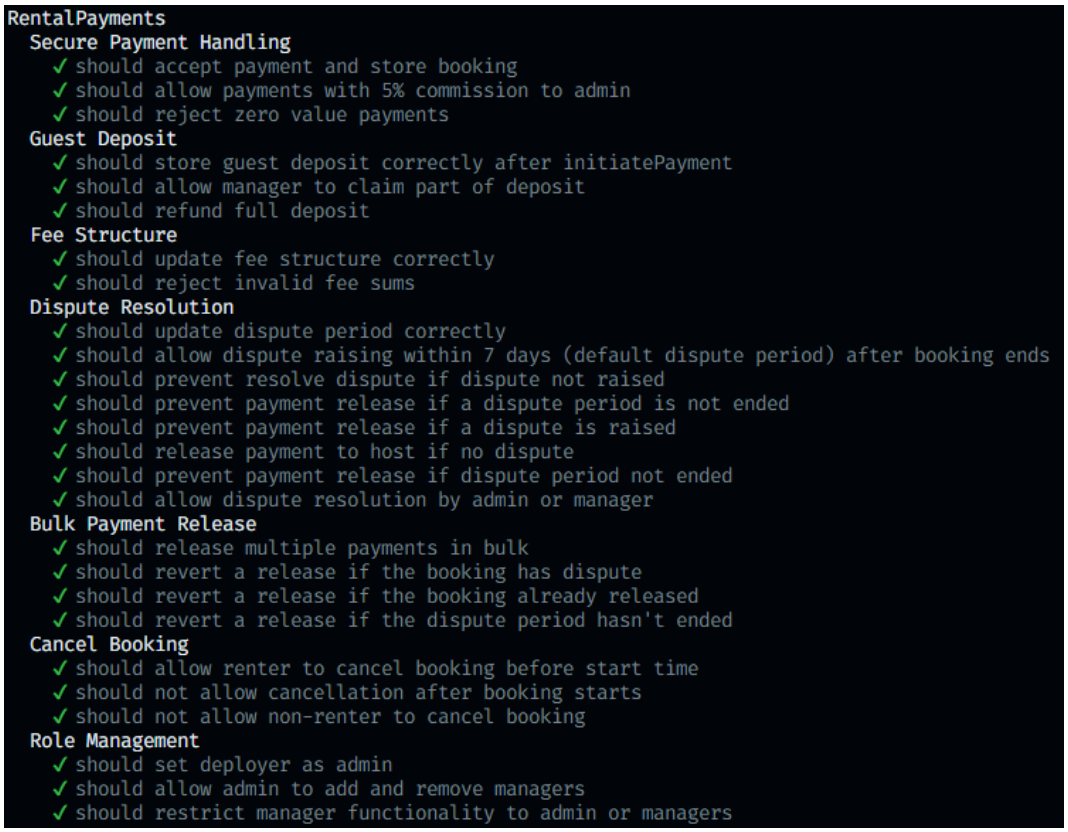
Tabel V.12 Hasil pengujian FT-09

Pre-kondisi	Prosedur pengujian	Input	Hasil yang diharapkan	Hasil yang didapatkan	Hasil
Sudah ada dispute yang perlu di-*review*.	1. Membuka halaman *admin* bagian *dispute*; 2. Menekan tombol "Favor Host" atau "Favor Renter"; 3. Melengkapi *dispute resolution form*.	**reason:** Test	a. Tombol "Confirm Resolution" dapat ditekan; b. Melanjutkan ke konfirmasi transaksi pada MetaMask *wallet*.	a. Tombol "Confirm Resolution" dapat ditekan; b. Melanjutkan ke konfirmasi transaksi pada MetaMask *wallet*.	✓
		-	Tombol "Confirm Resolution" tidak dapat ditekan.	Tombol "Confirm Resolution" tidak dapat ditekan.	✓

V.3.10 Hasil Pengujian UT-01

Gambar V.1 menunjukkan hasil *unit testing* pada UT-01. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa *smart contract* mampu menangani berbagai fitur utama yang diperlukan dalam sistem pemesanan hotel, yang meliputi pembayaran, pembatalan pemesanan, serta pengajuan dan penyelesaian sengketa.

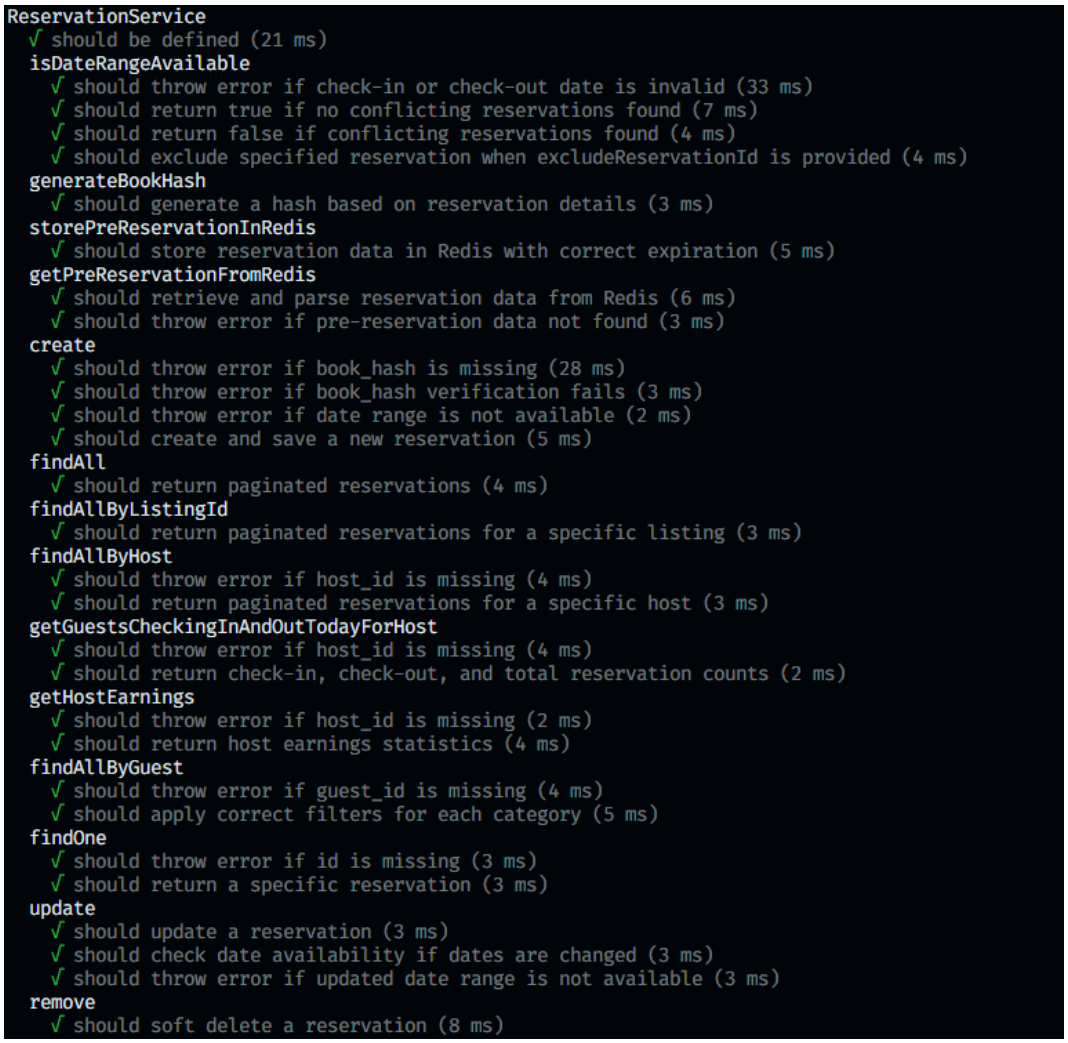
Berdasarkan *gas report* saat melakukan *unit testing*, rata-rata konsumsi *gas* untuk fungsi penting seperti `initiatePayment` (197,881 *gas*), `releasePayment` (79,543 *gas*), dan `resolveDispute` (60,428 *gas*) berada jauh di bawah batas blok Ethereum (30 juta *gas*). Hal ini menunjukkan bahwa transaksi dapat dilakukan dalam satu blok tanpa risiko gagal karena kehabisan *gas*. Selain itu, kontrak RentalPayments hanya menggunakan 10.4% dari *block gas limit*, yang berarti kontrak bisa di-*deploy* dengan efisien, tidak boros biaya, dan *scalable*.



Gambar V.1 Hasil pengujian UT-01

V.3.11 Hasil Pengujian UT-02

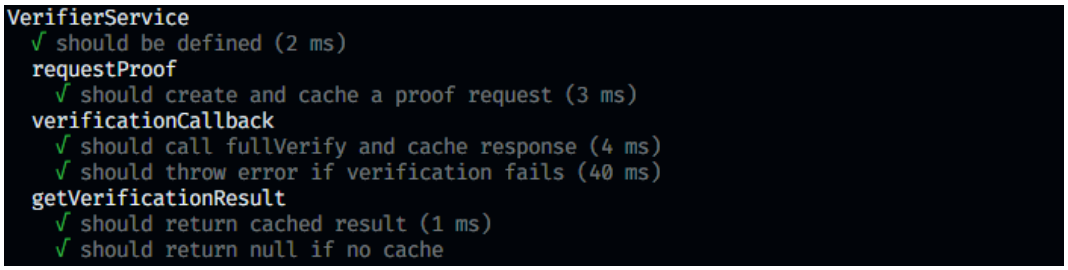
Gambar V.2 menunjukkan hasil *unit testing* pada UT-02. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa *backend* mampu menangani berbagai fitur utama yang diperlukan dalam sistem pemesanan hotel, yang meliputi pembuatan data *pre-reservation*, *booking hash generation*, serta CRUD untuk data reservasi pada *centralized database*.



Gambar V.2 Hasil pengujian UT-02

V.3.12 Hasil Pengujian UT-03

Gambar V.3 menunjukkan hasil *unit testing* pada UT-03. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa *backend* sebagai *verifier* mampu menangani verifikasi *proof* dari pengguna.

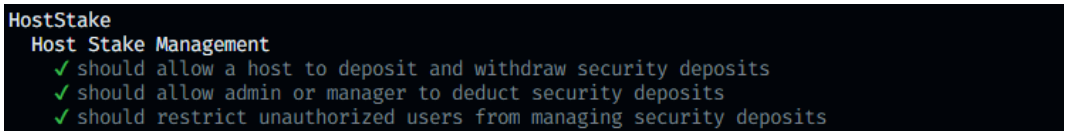


Gambar V.3 Hasil pengujian UT-03

V.3.13 Hasil Pengujian UT-04

Gambar V.4 menunjukkan hasil *unit testing* pada UT-04. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa *smart contract* mampu menangani berbagai fitur penting dalam sistem *staking* untuk *host*, yang meliputi fitur *deposit*, serta fitur *withdraw* untuk *host* dan *admin*.

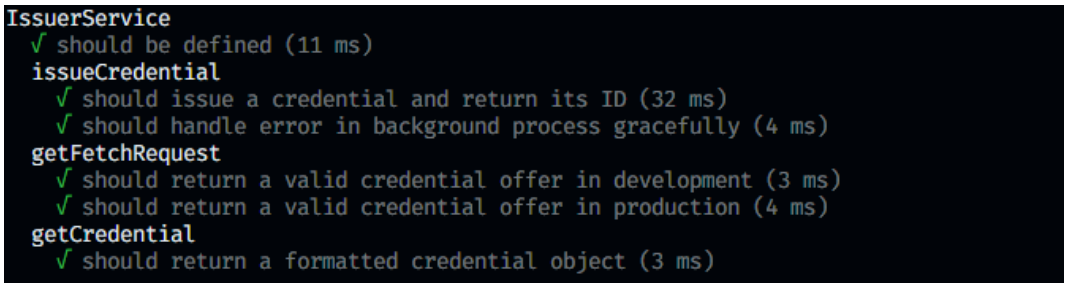
Berdasarkan *gas report* saat melakukan *unit testing*, rata-rata konsumsi *gas* untuk seluruh fungsi berada jauh di bawah batas blok Ethereum. Hal ini menunjukkan bahwa transaksi dapat dilakukan dalam satu blok tanpa risiko gagal karena kehabisan *gas*. Selain itu, kontrak HostStake hanya menggunakan 3.8% dari *block gas limit*, yang berarti kontrak bisa di-*deploy* dengan efisien, tidak boros biaya, dan *scalable*.



Gambar V.4 Hasil pengujian UT-04

V.3.14 Hasil Pengujian UT-05

Gambar V.5 menunjukkan hasil *unit testing* pada UT-05. Sistem berhasil melewati seluruh skenario pengujian yang telah didefinisikan. Dengan demikian, dapat disimpulkan bahwa *backend* sebagai *issuer* mampu menangani proses *crededential issuance* untuk pengguna (*holder*).



Gambar V.5 Hasil pengujian UT-05

V.3.15 Hasil Audit SA-01

Hasil *static analysis* menunjukkan beberapa potensi kerentanan keamanan dan inefisiensi pada kedua *smart contract*, yaitu:

a. Tidak disarankan menggunakan `block.timestamp` untuk komparasi;
b. Perbedaan versi Solidity yang digunakan pada smart contract yang dibuat dengan *smart contract* eksternal yang digunakan (`ReentrancyGuard.sol`);
c. *External call* sebelum perubahan *state* atau *events* setelah pemanggilan `transfer()`;
d. *Warning* untuk beberapa penamaan variabel yang tidak sesuai *naming convention*;
e. *Admin* seharusnya *immutable*.

Berdasarkan hasil *static analysis*, dilakukan beberapa perbaikan pada *smart contract* untuk meningkatkan keamanan dan efisiensi. Namun, beberapa temuan seperti penggunaan `block.timestamp` serta perbedaan versi Solidity tidak ditindaklanjuti lebih lanjut karena dinilai tidak berdampak signifikan terhadap *use case* yang bersifat terbatas dan tidak sensitif terhadap manipulasi waktu atau kompatibilitas lintas versi.

V.4 Diskusi

Berdasarkan hasil evaluasi yang telah dilakukan, sistem P2P AP telah berhasil memenuhi sebagian besar tujuan yang telah ditetapkan. Pengujian fungsional menunjukkan bahwa seluruh alur utama telah berjalan sesuai ekspektasi. Hal ini

menegaskan bahwa sistem mampu menjalankan transaksi secara otomatis tanpa keterlibatan pihak ketiga, serta menjaga privasi pengguna dengan tidak menyimpan data identitas dalam sistem. Selain itu, konsumsi *gas* untuk setiap fungsi tercatat efisien dan jauh di bawah batas maksimum *gas* Ethereum, yang memperkuat argumen bahwa sistem ini layak dioperasikan pada jaringan *blockchain* publik.

Hasil *static analysis* terhadap *smart contract* juga mendukung validitas desain sistem. Tidak ditemukan kerentanan keamanan kritis seperti *reentrancy* atau *overflow*, dan temuan minor telah ditindaklanjuti melalui perbaikan kode, meskipun beberapa temuan ada yang diabaikan karena tidak berpengaruh secara signifikan. Evaluasi ini menunjukkan bahwa *smart contract* telah dirancang dengan mempertimbangkan keamanan dan efisiensi.

Selain itu, walaupun aspek *usability* tidak menjadi fokus dalam penelitian ini, keterbatasan pada sisi *user experience* tetap menjadi catatan penting. Saat melakukan evaluasi, ditemukan bahwa antarmuka pengguna masih belum optimal, khususnya dalam hal kejelasan notifikasi transaksi dan penanganan kesalahan. Kondisi ini dapat memengaruhi kenyamanan dan kepercayaan pengguna dalam berinteraksi dengan sistem. Oleh karena itu, meskipun sistem telah berhasil secara fungsional dan teknis sebagai *proof of concept* dari platform pemesanan anonim berbasis *blockchain*, pengembangan lanjutan tetap diperlukan, khususnya pada peningkatan pengalaman pengguna, agar sistem ini dapat diadopsi secara lebih luas dalam konteks implementasi di dunia nyata.

BAB VI

KESIMPULAN DAN SARAN

VI.1 Kesimpulan

Berdasarkan pelaksanaan dan evaluasi yang telah dilakukan, dapat disimpulkan bahwa seluruh rumusan masalah telah berhasil dijawab dan tujuan penelitian telah tercapai. Kesimpulan dari hasil evaluasi terhadap sistem yang dikembangkan adalah sebagai berikut:

1. Sistem berhasil mengimplementasikan *smart contract* pada *peer-to-peer accommodation platform* (P2P AP) untuk mengelola transaksi antara *guest* dan *host* secara otomatis. *Smart contract* diimplementasikan dengan cara merancang skema transaksi *on-chain* yang mencakup proses pembayaran, pembatalan, klaim deposit, dan penyelesaian sengketa tanpa keterlibatan pihak ketiga. Hasil evaluasi menunjukkan bahwa *smart contract* mampu mengeksekusi seluruh proses tersebut secara mandiri, sehingga meningkatkan transparansi, keamanan, dan kepercayaan dalam transaksi.
2. Sistem telah merancang dan mengimplementasikan proses pemesanan anonim menggunakan *booking credential* berbasis *self-sovereign identity* dan *zero-knowledge proof*. Proses ini diimplementasikan dengan memanfaatkan SDK dari Privado ID untuk menerbitkan *credential* yang dapat diverifikasi oleh *host*. Hasil evaluasi menunjukkan bahwa *credential* berhasil diterbitkan kepada pengguna dan dapat diverifikasi oleh *host* tanpa mengungkap data identitas, menjawab tantangan privasi dalam sistem pemesanan *online*.
3. Sistem verifikasi pengguna berhasil dikembangkan untuk mencegah penyalahgunaan dalam sistem anonim tanpa mengakses atau menyimpan data identitas pengguna. Hal ini dicapai dengan mengintegrasikan *proof of uniqueness*, *proof of liveness*, dan *proof of identity* menggunakan *verifiable*

credentials dari pihak ketiga. Hasil evaluasi menunjukkan sistem mampu mencegah pembuatan multi-akun sambil tetap menjaga privasi pengguna.

Secara keseluruhan, sistem yang dibangun telah divalidasi melalui serangkaian pengujian, termasuk pengujian fungsional, pengujian *gas* pada *smart contract*, serta evaluasi terhadap mekanisme SSI dan ZKProof. Hasil dari seluruh pengujian tersebut menunjukkan bahwa sistem mampu berjalan sesuai dengan rancangan dan memenuhi kebutuhan yang telah ditetapkan. Dengan demikian, dapat disimpulkan bahwa masalah utama yang dirumuskan di Bab I berhasil diselesaikan, dan seluruh tujuan penelitian telah tercapai.

VI.2 Saran

Sebagai lanjutan dari penelitian ini, terdapat beberapa saran pengembangan yang dapat dilakukan untuk menyempurnakan sistem, terutama dalam menghadapi kondisi nyata dan skenario *edge-case* yang mungkin belum sepenuhnya terakomodasi:

1. Untuk mencegah penyalahgunaan oleh aktor jahat dalam sistem anonim, dapat dikembangkan fitur reputasi berbasis *verifiable credential*. Misalnya, *host* atau *guest* yang menyelesaikan transaksi tanpa sengketa dapat menerima *reputation badge* yang diverifikasi dan dapat ditunjukkan saat memesan atau menerima tamu. Hal ini tetap menjaga anonimitas namun memberi insentif bagi perilaku baik.
2. Untuk mencegah spam *booking* dari *guest* yang sama, tanpa melanggar prinsip anonimitas, dapat dibuat sistem *anonymous token rate-limiting*. *Guest* yang melakukan spam akan terdeteksi melalui *nullifier* dan dicegah memesan ulang selama periode tertentu.
3. Untuk mencegah *host* palsu yang mendaftarkan lokasi fiktif atau properti yang sama berulang kali, dapat dikembangkan sistem berbasis *geohash* atau integrasi dengan *oracle* lokasi. Misalnya, *host* perlu melakukan *proof of presence* di lokasi tertentu secara anonim.

DAFTAR PUSTAKA

Agag, Gomaa, dan Riyad Eid. 2019. "Examining the antecedents and consequences of trust in the context of peer-to-peer accommodation." *International Journal of Hospitality Management* 81 (Agustus):180–92. https://doi.org/10.1016/j.ijhm.2019.04.021.

Antonopoulos, Andreas M, dan Gavin D Wood Ph. 2019. *Mastering Ethereum*. The Ethereum Book LLC.

Basha, Shaik Johny, Venkata Srinivasu Veesam, Tamminina Ammannamma, Sirisha Navudu, dan M. V.V.S. Subrahmanyam. 2021. "Security enhancement of digital signatures for blockchain using EdDSA algorithm." Dalam *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, 274–78. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICICV50876.2021.9388411.

Bhushan, Bharat, Preeti Sinha, K. Martin Sagayam, dan Andrew J. 2021. "Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions." *Computers and Electrical Engineering* 90 (Maret). https://doi.org/10.1016/j.compeleceng.2020.106897.

Blengini, Isabella, dan Beatrice Venturini. 2024. "Transparency & conflict resolution in Airbnb & other two-sided markets." EHL Insights. 14 Maret 2024. https://hospitalityinsights.ehl.edu/transparency-conflict-resolution-airbnb-two-sided-markets.

Brown, Tim, dan Colin Funk. 2008. "Design Thinking." www.hbr.org.

Buterin, Vitalik. 2014. "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform."

Česnuitytė, Vida, Andrzej Klimczuk, Cristina Miguel, dan Gabriela Avram. 2022. *The Sharing Economy in Europe: Developments, Practices, and Contradictions*. *The Sharing Economy in Europe: Developments, Practices, and Contradictions*. Springer International Publishing. https://doi.org/10.1007/978-3-030-86897-0.

Chen, Thomas, Hui Lu, Teeramet Kunpittaya, dan Alan Luo. 2022. "A Review of zk-SNARKs," Februari. http://arxiv.org/abs/2202.06877.

cheqd. 2023. " Self-sovereign identity explained." https://cheqd.io/ssi/.

Dahlberg, Rasmus, Tobias Pulls, dan Roel Peeters. 2016. "Efficient Sparse Merkle Trees Caching Strategies and Secure (Non-)Membership Proofs." Dalam , 199–215. https://doi.org/https://doi.org/10.1007/978-3-319-47560-8_13.

Dann, David, Christian Peukert, Carl Martin, Christof Weinhardt, dan Florian Hawlitschek. 2020. "Blockchain and Trust in the Platform Economy: The Case of Peer-to-Peer Sharing." Dalam *WI2020 Zentrale Tracks*, 1459–73. GITO Verlag. https://doi.org/10.30844/wi_2020_n2-dann.

Dieye, Mohameden, Pierre Valiorgue, Jean Patrick Gelas, El Hacen Diallo, Parisa Ghodous, Frederique Biennier, dan Eric Peyrol. 2023. "A Self-Sovereign Identity Based on Zero-Knowledge Proof and Blockchain." *IEEE Access* 11:49445–55. https://doi.org/10.1109/ACCESS.2023.3268768.

Ethereum.org. 2023. "Ethereum Development Documentation." 2023. https://ethereum.org/en/developers/docs/.

European Commission. 2025. "eIDAS Regulation." 5 Mei 2025. https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation.

Gan, Qing Qiu, dan Raymond Yiu Keung Lau. 2024. "Trust in a 'trust-free' system: Blockchain acceptance in the banking and finance sector."

Technological Forecasting and Social Change 199 (Februari). https://doi.org/10.1016/j.techfore.2023.123050.

GeeksforGeeks. 2022. "Socio-technical Systems." GeeksforGeeks. 28 November 2022. https://www.geeksforgeeks.org/socio-technical-systems/.

Hawlitschek, Florian, Benedikt Notheisen, dan Timm Teubner. 2018. "The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy." *Electronic Commerce Research and Applications* 29 (Mei):50–63. https://doi.org/10.1016/j.elerap.2018.03.005.

Huang, Ying-Kai. 2021. "Attribution Bias on Online Reputation Systems." *Social Science Research Network*, April. https://ssrn.com/abstract=3834091.

Iden3. 2023. "Iden3 Documentation." 2023. https://docs.iden3.io/.

Intersoft Consulting. t.t. "General Data Protection Regulation." Diakses 10 Mei 2025. https://gdpr-info.eu/.

Kanani, Jaynti, Sandeep Nailwal, dan Anurag Arjun. 2019. "Matic Whitepaper," April.

Lux, Andras Zolt´an, Dirk Thatmann, Sebastian Zickau, dan Felix Beierle. 2020. *Distributed-Ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials*. IEEE.

Malcolm, Jess. 2024. "Exact steps you need to take now as Fidelity confirms 77,000 customers' personal information exposed in data breach." *The Sun*, 11 Oktober 2024. https://www.the-sun.com/money/12654629/fidelity-cybersecurity-hacking-data-stolen-tips/.

Mayer, Roger C, James H Davis, dan F David Schoorman. 1995. “An Integrative Model of Organizational Trust.” *Source: The Academy of Management Review*. Vol. 20.

Mittal, Pulkit. 2025. “Privado ID Documentation.” Privado ID Documentation. 23 Januari 2025. https://docs.privado.id/.

Moya, Cristina Vilchez, Juan Ramón Bermejo Higuera, Javier Bermejo Higuera, dan Juan Antonio Sicilia Montalvo. 2023. “Implementation and Security Test of Zero-Knowledge Protocols on SSI Blockchain.” *Applied Sciences (Switzerland)* 13 (9). https://doi.org/10.3390/app13095552.

Nakamoto, Satoshi. 2008. “Bitcoin: A Peer-to-Peer Electronic Cash System.” www.bitcoin.org.

Oliveira, Tiago, Matilde Alhinho, Paulo Rita, dan Gurpreet Dhillon. 2017. “Modelling and testing consumer trust dimensions in e-commerce.” *Computers in Human Behavior* 71 (Juni):153–64. https://doi.org/10.1016/j.chb.2017.01.050.

Peffers, Ken, Tuure Tuunanen, Marcus A. Rothenberger, dan Samir Chatterjee. 2007. “A Design Science Research Methodology for Information Systems Research.” *Journal of Management Information Systems* 24 (3): 45–77. https://doi.org/10.2753/MIS0742-1222240302.

Rabbani, Maheswara, Juan Daniel Wijaya, Rendy Sanjaya Kusuma, Wilhelmus Billion Pius Purba, dan Robert Marchelino Tajib. 2023. “Digital Payments in Indonesia: Understanding the Effect of Application Security on User Trust.” *Indonesian Journal of Computer Science Attribution* 12 (5): 2475.

Raipurkar, Abhijeet R., Shreyas Bobde, Anurag Tripahi, dan Mohit Sahu. 2023. “Digital Identity System Using Blockchain-based Self Sovereign Identity & Zero Knowledge Proof.” Dalam *OCIT 2023 - 21st International Conference on Information Technology, Proceedings*, 611–

16. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/OCIT59427.2023.10430981.

Raj, Koshik. 2019. *Foundations of Blockchain The pathway to cryptocurrencies and decentralized blockchain applications*. Packt Publishing.

Reed, Drummond, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, Markus Sabadello, dan Jonathan Holt. 2020. "Decentralized Identifiers (DIDs) v1.0." https://www.w3.org/TR/2020/WD-did-core-20201001/.

Rosoon, Yuwana, Chidchanok Choksuchat, dan Pattara Aiyarak. 2023. "Decentralized Trusted Database Approach to Online Product Reviews." Dalam *2023 15th International Conference on Information Technology and Electrical Engineering, ICITEE 2023*, 282–86. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICITEE59582.2023.10317755.

Rundle, James. 2024. "Fintech Company Finastra, Used by the Largest Banks, Discloses Hack." *The Wall Street Journal*, 21 November 2024. https://www.wsj.com/articles/fintech-company-finastra-used-by-the-largest-banks-discloses-hack-ef5a575d.

Solidity. 2024. "Solidity Documentation Release 0.8.29 Ethereum." https://docs.soliditylang.org/_/downloads/en/latest/pdf/.

Sporny, Manu, Dave Longley, dan David Chadwick. 2022. "Verifiable Credentials Data Model v1.1." World Wide Web Consortium (W3C). 3 Maret 2022. Verifiable Credentials Data Model v1.1.

The Times. 2024. "Get ready for your own CrowdStrike, City regulator tells firms," 31 Oktober 2024. https://www.thetimes.com/business-money/companies/article/get-ready-for-your-own-crowdstrike-city-regulator-tells-firms-tp0t57pst.

Tushev, Miroslav, Fahimeh Ebrahimi, dan Anas Mahmoud. 2022. "A Systematic Literature Review of Anti-Discrimination Design Strategies in the Digital Sharing Economy." *IEEE Transactions on Software Engineering* 48 (12): 5148–57. https://doi.org/10.1109/TSE.2021.3139961.

Tussyadiah, Iis P., dan Sangwon Park. 2018. "When guests trust hosts for their words: Host description and trust in sharing economy." *Tourism Management* 67 (Agustus):261–72. https://doi.org/10.1016/j.tourman.2018.02.002.

Veldhuizen, F L M. 2020. "How to Design a Good Reputation System for an Online Peer-to-Peer Platform." University of Twente.

Venable, John R, Jan Pries-Heje, dan Richard L Baskerville. 2017. "Choosing a Design Science Research Methodology." *Australia Choosing a Design Science Research Methodology*. Vol. 2. https://aisel.aisnet.org/acis2017/112.

Wang, Fennie, dan Primavera De Filippi. 2019. "Self-Sovereign Identity in a Globalized World: Credentials-Based Identity Systems as a Driver for Economic Inclusion." *Frontiers in Blockchain* 2. https://doi.org/10.3389/fbloc.2019.00028.

Whitehat, Barry, Jordi Baylina, dan Marta Bellés. 2019. "Baby Jubjub Elliptic Curve."

Wood, Gavin. 2022. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." *cryptodeep.ru*, Oktober.

Xu, Jie, Kim Le, Annika Deitermann, dan Enid Montague. 2014. "How different types of users develop trust in technology: A qualitative analysis of the antecedents of active and passive user trust in a shared technology." *Applied Ergonomics* 45 (6): 1495–1503. https://doi.org/10.1016/j.apergo.2014.04.012.

Zhu, Yuxin, Dazuo Tian, dan Feng Yan. 2020. “Effectiveness of Entropy Weight Method in Decision-Making.” *Mathematical Problems in Engineering* 2020. https://doi.org/10.1155/2020/3564835.

Zloteanu, Mircea, Nigel Harvey, David Tuckett, dan Giacomo Livan. 2021. “Judgments in the Sharing Economy: The Effect of User-Generated Trust and Reputation Information on Decision-Making Accuracy and Bias.” *Frontiers in Psychology* 12 (November). https://doi.org/10.3389/fpsyg.2021.776999.

LAMPIRAN A
REPOSITORI GITHUB

A.1. Repositori *Frontend*

Repositori ini berisi *source code* untuk antarmuka pengguna (*frontend*) dari sistem P2P AP. Aplikasi *frontend* berfungsi untuk menghubungkan pengguna (*guest* dan *host*) dengan layanan *backend* dan *smart contract* di *blockchain*. Semua tampilan, interaksi pengguna, dan pemrosesan input pengguna dikembangkan dan dikelola di dalam repositori ini.

URL: https://github.com/ceavinrufus/TA-frontend

A.2. Repositori *Backend*

Repositori *backend* berisi logika *server-side* dan layanan API yang mengatur komunikasi antara *frontend*, *database*, dan *smart contract*. *Backend* ini bertanggung jawab atas autentikasi pengguna, manajemen data pemesanan, serta penyimpanan informasi yang diperlukan untuk proses transaksi.

URL: https://github.com/ceavinrufus/TA-backend

A.3. Repositori *Smart Contract*

Repositori ini menyimpan kumpulan *smart contract* yang digunakan untuk mengatur alur transaksi utama dalam platform P2P AP, seperti pemesanan, penyelesaian sengketa, dan jaminan keamanan.

URL: https://github.com/ceavinrufus/TA-smart-contract

A.4. Repositori *Indexer*

Repositori *indexer* berisi konfigurasi dan kode yang digunakan untuk membangun *subgraph* dengan menggunakan The Graph Protocol. *Subgraph* ini memungkinkan data dari *smart contract* di-*query* secara efisien menggunakan GraphQL, sehingga

memudahkan *frontend* dan *backend* dalam mengakses informasi *on-chain* secara *real-time*. Repositori ini sangat penting dalam mendukung visibilitas data dan performa sistem.

URL: https://github.com/ceavinrufus/TA-subgraph

LAMPIRAN B
WAWANCARA PENGGUNA

B.1 Wawancara Penyewa (*Guest*)

B.1.1 *Guest* 1 (Nico)

Umur: 27 tahun

Pekerjaan: Desainer grafis *freelance*

Asal: Kenya

Kebiasaan *booking*: Sering menyewa akomodasi saat bepergian untuk kerja *remote*

Q&A:

- Q: *Can you tell me what's usually on your mind when you're about to book a place on a P2P platform?*
- A: *Hmm… I mostly check the pictures and reviews. But yeah, sometimes I still feel unsure. The listing looks great, but you never know if it's real until you're there*.
- Q: *Do you feel the information about the host is clear enough?*
- A: *Not always. Some profiles barely have any info, and I can't really tell who I'll be dealing with. It's kind of a gamble*.
- Q: *How do you feel about giving your personal data during booking?*
- A: *Honestly? I don't like it. I mean, why do they need my ID and all that just to rent a room? I'm not even sure who's accessing it*.
- Q: *Have you ever faced payment issues?*
- A: *One time, yes. My booking got canceled, and I had to wait almost a week to get my money back*.
- Q: *Do you trust the reviews on those platforms?*
- A: *I did, but then I heard some platforms remove negative ones. That's messed up. It makes you wonder how many bad experiences are hidden*.

- Q: *Ever felt judged or rejected unfairly when booking?*
- A: *Hmm… I don't know for sure, but once I used a photo of me wearing a hoodie, and my request got declined twice. After I changed the photo, suddenly accepted. So yeah… felt weird*.
- Q: *What's the worst experience you had as a P2P accommodation platform user?*
- A: *Besides the one I mentioned before, there's nothing else.*
- Q: *Would you try a new platform that promises more transparency and fairness?*
- A: *Yeah, I'd be open to trying something new, especially if it fixed the pain points I've had before. But I'd still be cautious in the beginning.*

B.1.2 *Guest* 2 (Dea)

Umur: 34 tahun

Pekerjaan: *Blockchain developer*

Asal: Kenya

Kebiasaan *booking*: Liburan keluarga dan *business trip*

Q&A:

- Q: *What's your general experience using P2P accommodation platforms?*
- A: *It's mostly fine, but sometimes the place is not like the photos. That's frustrating when you're traveling with kids*.
- Q: *How do you feel about submitting your personal information on those platforms?*
- A: *I don't feel great about it. I usually do it because I have no choice, and I just hope that my data is safe*.
- Q: *Ever had issues with the payment process?*
- A: *Not really, except that they only accept credit cards or e-wallets*. *I wish it can be more flexible*.

- Q: *Do you think platforms are transparent with user reviews?*
- A: *No. I've seen cases where bad reviews disappear, although I believe they have their own reasons. It's unfair, and it makes it hard to trust*.
- Q: *Any issues with discrimination?*
- A: *Yes and no. I haven't got any issues. However, I think sometimes they choose who they rent to based on names or pictures, because my friend had that happen more than once*.
- Q: *What's the worst experience you had as a P2P accommodation platform user?*
- A: *Nothing, so far so good, nothing significant.*
- Q: *Would you try a new platform that promises more transparency and fairness?*
- A: *That would be great! I am ready to be the early adopters.*

B.1.3 *Guest* 3 (Rian)

Umur: 22 tahun

Pekerjaan: Mahasiswa

Asal: Amerika Serikat

Kebiasaan *booking*: Sering traveling hemat dan *last minute*

Q&A:

- Q: *Do you usually trust listings on booking platforms?*
- A: *Mostly yes, but sometimes what you see online is totally different from reality*. *But it's not a big problem for me.*
- Q: *Are you okay with giving your personal data?*
- A: *Since you said I should assume this is a new platform, I will avoid it if I can. Because I don't even know where the data goes. Too risky for me.*
- Q: *Have you faced payment problems?*

- A: *Yes. Payment went through, but the booking didn't. Took three days to fix. But it's kind of one-time issue. I never experienced it after that.*
- Q: *Do you feel you're treated fairly as a guest?*
- A: *I've had my booking declined even though everything looked fine. No reason given. It felt unfair.*
- Q: *Do you trust the reviews on those platforms?*
- A: *Yes, I do. I mean, I don't think there's a better way to know the host's reputation than through that.*
- Q: *There's a news report about some booking platforms selling user data. How do you feel about that?*
- A: *Honestly, that's exactly what I'm afraid of. I feel like once your data is out, there's no turning back. That's why I hesitate to sign up on new or unfamiliar platforms.*
- Q: *Would you try a new platform that promises more transparency and fairness?*
- A: *Yes, I'm always open with new options, especially if it gives a unique value for me.*

B.1.4 Guest 4 (Arif)

Umur: 31 tahun

Pekerjaan: *Freelance* videografer

Asal: Indonesia

Kebiasaan Booking: Sering bepergian untuk proyek dan *shooting* luar kota

Q&A:

- Q: Mas, biasanya hal apa yang paling Mas perhatikan sebelum *booking* penginapan lewat platform?
- A: Yang penting harganya masuk akal, terus lokasinya sesuai. Kalo itu udah oke, biasanya lanjut aja.

- Q: Mas merasa aman gak waktu diminta isi data pribadi pas daftar atau *booking*?
- A: Gak terlalu mikirin sih. Selama ada *terms and conditions* yang jelas dan *platform*-nya kelihatan profesional, ya udah, saya jalanin aja.
- Q: Jadi Mas gak keberatan data pribadi disimpan sama platform?
- A: Nggak juga. Emang udah biasa isi data kalau mau pesan apa-apa online. Lagian saya anggap wajar aja. Walaupun tetap harus hati-hati, sih. Soalnya sekarang banyak kasus data bocor.
- Q: Tapi kalau penyedia platform jual data Mas seenaknya, gimana?
- A: Ya gak mau, lah. Makanya *terms and conditions*-nya harus jelas.
- Q: Okedeh, lanjut. Ada pengalaman buruk soal pembayaran atau transparansi ulasan gak?
- A: Pembayaran sih sejauh ini aman. Tapi soal *review* ya kadang curiga juga, soalnya semua bintang lima, gak masuk akal sih.
- Q: Pernah merasa gak nyaman karena ditolak atau dinilai dari penampilan/foto?
- A: Belum, sih. Mungkin karena beruntung aja gak ketemu sama *host* yang rasis.
- Q: Tapi Mas pernah dengar ada isu pemesanan hotel ditolak karena penampilan/foto gak?
- A: Gak pernah dengar, sih.
- Q: Gimana kalau ada platform baru yang nggak simpan banyak data atau bisa *booking* tanpa kasih identitas asli?
- A: Menarik untuk dicoba, tetapi sejauh ini *prefer* yang udah pasti aja.

B.1.5 *Guest* 5 (Jajang)

Umur: 24 tahun

Pekerjaan: *Software engineer*

Asal: Indonesia

Kebiasaan Booking: *Remote work stay*

Q&A:

- Q: Biasanya apa yang Mas Jajang cari sebelum *booking* tempat?
- A: WiFi *kenceng* dan tempatnya gak jauh dari *coffee shop*, *wkwk*. Tapi serius, *review* dan foto juga penting, sih.
- Q: Nah, kebetulan karena Mas mention tentang *review*, menurut Mas Jajang sistem ulasan di platform bisa dipercaya gak?
- A: Seringnya bisa, tapi saya juga tahu ada yang bisa diatur. Pernah nemu *listing* jelek tapi ulasannya bagus banget, itu udah *red flag* sih.
- Q: Mas merasa nyaman gak kalau harus kasih data pribadi ke platform?
- A: Gapapa sih, toh data kita udah jadi *open source* sekarang, hahaha. *Jokes aside*, jujur males kalau platformnya belum dikenal. Apalagi kalau gak jelas siapa yang pegang data kita. Kalau udah familiar kayak Airbnb ya beda cerita.
- Q: Gimana soal pengalaman diskriminasi atau penolakan *booking* tanpa alasan jelas? Mas Jajang pernah ada pengalaman begitu, gak?
- A: Belum pernah, tapi saya pernah bantuin temen *booking* dan ditolak. Bukannya mau *negative thinking*, waktu itu *booking*-nya di luar negeri, dan nama dia islam banget, jadi ya, agak mikir juga sih alasannya.
- Q: Gimana kalau ada platform baru yang nggak simpan banyak data atau bisa *booking* tanpa kasih identitas asli?
- A: Wah, itu menarik banget. Kalau tetap bisa aman dan *host* juga merasa nyaman, saya mau coba.

B.2 Wawancara Pemilik Properti (*Host*)

B.2.1 *Host* 1 (Eva)

Umur: 36 tahun

Pekerjaan: Ibu rumah tangga

Penggunaan platform: Menyewakan rumah di kota wisata

Q&A:

- Q: *What's your biggest concern when hosting?*
- A: *I'm always worried guests might break something or make a mess. The platform doesn't always help.*
- Q: *Do you get enough info about your guests?*
- A: *Not really. I just see a name and profile picture. I wish I knew more before accepting a booking.*
- Q: *Do you trust the platform to protect your property?*
- A: *Not 100%. I've had to pay for repairs myself before.*
- Q: *Any payment issues from your side?*
- A: *Yes, sometimes payments are delayed. It messes up my budgeting.*
- Q: *What if your account gets suspended without warning?*
- A: *That would be terrible. I've heard stories like that, and it makes me anxious.*
- Q: *Are you comfortable with how much data the platform collects from you?*
- A: *Honestly, not really. I feel like they, not all platform, ask for a lot.*
- Q: *Would you be okay if the platform shares your data with third parties?*
- A: *No way! That should be my choice. I use the platform to rent my place, not to have my data sold around.*

B.2.2 *Host* 2 (Tono)

Umur: 32 tahun

Pekerjaan: Pensiunan

Penggunaan Platform: Menyewakan villa keluarga

Q&A:

- Q: *How do you feel about your guests' data?*
- A: *I wish we could know more about them, just to be safe.*

- Q: *Any experience with problematic guests?*
- A: *Yes, once. Noisy and broke some items. The platform didn't really support me.*
- Q: *How's the payment process on the platform you use?*
- A: *It's okay most of the time.*
- Q: *Are you comfortable with how much data the platform collects from you?*
- A: *I mean, I have no choice but to trust them with my data. However, if the platform is a new one, with no reputation yet, I am not comfortable with that. I am afraid the platform will do something terrible with my data.*
- Q: *If there is a new platform that offered better privacy controls and transparency, would you consider switching?*
- A: *For sure. If I felt safer and more respected as a host, I'd definitely try something new.*

B.2.3 *Host* 3 (Lisa)

Umur: 29 tahun

Pekerjaan: Freelancer

Penggunaan Platform: Menyewakan apartemen di pusat kota

Q&A:

- Q: *What makes you hesitate to accept a booking?*
- A: *When the guest is new and has no reviews. There's just no way to check if they're legit.*
- Q: *What's your biggest worry when hosting?*
- A: *Safety. Not just for the property, but also for my neighborhood. If something goes wrong, I'm the one who has to deal with it. That's why I should know the guest.*
- Q: *Have you had issues with the platform's rules?*
- A: *Yes. They once changed the terms without notice. Suddenly, my listing got hidden.*

- Q: *How's the payment process on the platform you use?*
- A: *I have no problem regarding that, fortunately.*
- Q: *How do you feel about sharing your own personal data with the platform?*
- A: *I don't like it, but I don't have a choice. Some platform require ID, tax info, phone numbers… but I don't know how it's stored or who can see it.*
- Q: *Have you ever felt your data was used in a way you didn't agree with?*
- A: *I can't say for sure, but I've received strange marketing emails that make me wonder if my data was shared.*

B.3 Platform Operator

Jabatan: CEO *startup* yang bergerak di bidang P2P AP

Asal: Cina

Peran: Mengawasi keseluruhan sistem

Q&A:

- Q: *What's your biggest concern as a platform operator?*
- A: *Trust. Everything depends on it, especially for P2P platform. If users stop trusting our platform, we're done. That includes trust in our policies, our fairness, and most importantly, how we handle their data.*
- Q: *What's your biggest operational challenge, especially regarding data privacy?*
- A: *Because we store user data in our database, if there's a breach, it affects everyone. It's a big responsibility.*
- Q: *How do users usually react to changes in policy?*
- A: *Many get upset, especially when rules change suddenly or reviews get removed. They think we're biased, even when we're just following protocol.*
- Q: *Have you faced issues with third-party service providers?*
- A: *Yes. Once, a security flaw in our payment gateway exposed user data. Even though it wasn't our fault, we were the ones blamed.*

- Q: *Is it a reputable payment gateway provider?*
- A: *Yes, you must know this provider if you've built a SaaS product.*
- Q: *Do you change the provider after that?*
- A: *No, because I don't have any other options. However, I'm open to a new solutions.*
- Q: *Do you believe centralized data storage is sustainable in the long term?*
- A: *I'm starting to doubt it. It puts too much risk on us as the operator. I've been exploring decentralized alternatives, but they're not yet user-friendly enough for mass adoption.*
- Q: *How transparent is your platform about data handling to users?*
- A: *We publish a privacy policy, but let's be honest, not many users read it. We're working on a clearer, human-readable version to improve trust.*
- Q: *What's your view on letting users control their own data?*
- A: *I support it, conceptually. But technically, it's still challenging to implement without hurting UX or breaking integrations with partners.*
- Q: *Do you think users are ready for a system where they own their identity and credentials?*
- A: *Some are. Especially the privacy-conscious crowd. But most users still prioritize convenience. The tech needs to become invisible before it becomes mainstream.*