BAB III
METODE PENELITIAN

Penelitian ini menggunakan metode *Design and Development* (D&D). Menurut Richey dan Klein (2014, hlm. 22), metode ini termasuk penelitian pengembangan, yang mendokumentasikan tahapan *Instructional System Design* (ISD), mulai dari analisis kebutuhan, desain, pengembangan, hingga evaluasi formatif dari beberapa. Metode D&D adalah pendekatan penelitian yang berfokus pada perancangan, pengembangan, dan evaluasi produk atau model. Tujuan utama dari metode ini adalah untuk menciptakan solusi yang efektif dan praktis terhadap masalah yang ada melalui serangkaian tahapan yang terstruktur dan sistematis. Dalam konteks penelitian ini, metode D&D diterapkan untuk membangun sebuah sistem berbasis website yang mampu menerbitkan dan memverifikasi keaslian *e-certificate* menggunakan teknologi *blockchain*.

Metode *Design and Development* sering kali melibatkan siklus iteratif yang terdiri dari beberapa tahapan utama, yaitu analisis, desain, pengembangan, dan evaluasi. Setiap tahapan saling terkait dan memungkinkan adanya perbaikan berkelanjutan. Dengan menggunakan metode ini, peneliti tidak hanya menghasilkan sebuah produk, tetapi juga menguji dan mengevaluasi efektivitasnya secara mendalam. Dalam penelitian ini, tahapan metode D&D yang akan diterapkan adalah sebagai berikut:

1) Analisis: Tahap ini berfokus pada identifikasi kebutuhan sistem dan studi literatur untuk memahami masalah secara mendalam dan menentukan teknologi yang paling sesuai.
2) Desain: Tahap ini mencakup perancangan arsitektur sistem, antarmuka pengguna, dan logika *smart contract* yang akan diimplementasikan.
3) Pengembangan dan Implementasi: Tahap ini merupakan realisasi dari desain yang telah dibuat menjadi sebuah produk yang fungsional, yaitu website dan *smart contract*.

4) Evaluasi: Tahap ini bertujuan untuk menguji dan menilai efektivitas serta fungsionalitas sistem yang telah dikembangkan, termasuk pengujian keaslian verifikasi *e-certificate*.
5) Pelaporan: Tahap terakhir ini adalah mendokumentasikan seluruh proses penelitian, hasil yang didapat, serta kesimpulan dan saran untuk pengembangan lebih lanjut.

Dengan demikian, metode Design and Development sangat relevan dan sesuai untuk penelitian ini karena memungkinkan peneliti untuk tidak hanya mengimplementasikan teknologi *blockchain*, tetapi juga memastikan bahwa sistem yang dihasilkan benar-benar berfungsi dengan baik dan mampu menjawab permasalahan terkait keaslian *e-certificate*.

3.1. Analisis

Tahap analisis merupakan langkah awal dalam metode *Design and Development* (D&D) yang bertujuan untuk mengidentifikasi dan memahami permasalahan, mengumpulkan kebutuhan sistem, serta menentukan solusi yang paling sesuai. Pada tahap ini, peneliti melakukan studi mendalam untuk mendapatkan informasi yang dibutuhkan dalam perancangan dan pengembangan sistem. Analisis ini dibagi menjadi tiga sub-bagian utama: analisis kebutuhan sistem, studi literatur, dan analisis teknologi.

3.1.1. Analisis Kebutuhan Sistem

Analisis ini dilakukan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dari sistem yang akan dibangun. Tujuannya adalah untuk memastikan sistem dapat berjalan sesuai harapan pengguna dan memenuhi standar teknis yang telah ditentukan.

1) Kebutuhan Fungsional

a) Penerbitan E-Certificate**:** Sistem harus menyediakan antarmuka bagi User untuk mengunggah data sertifikat (nama, nomor sertifikat, acara, dll.) dan memprosesnya untuk disimpan di *blockchain*.

b) Penyimpanan Hash di *Blockchain***:** Sistem harus mampu mengonversi data sertifikat menjadi nilai hash, kemudian menyimpannya sebagai transaksi yang tidak dapat diubah di dalam *blockchain* melalui *smart contract*.

c) Verifikasi Keaslian**:** Sistem harus menyediakan fitur verifikasi yang memungkinkan pengguna (baik pemilik sertifikat maupun pihak ketiga) untuk memasukkan nomor sertifikat dan mengonfirmasi keasliannya dengan membandingkan hash yang ada di *blockchain*.

d) Akses Pengguna**:** Sistem harus membedakan hak akses antara User (untuk menerbitkan sertifikat) dan pengguna umum (untuk melihat dan memverifikasi sertifikat).

e) Tampilan Sertifikat**:** Sistem harus dapat menampilkan detail sertifikat yang sah setelah proses verifikasi berhasil.

2) Kebutuhan Non-Fungsional

a) Keamanan**:** Data sensitif harus dilindungi, dan interaksi dengan *blockchain* harus aman dari serangan siber.

b) Keterandalan**:** Sistem harus dapat diakses dan beroperasi dengan stabil 24/7.

c) Kemudahan Penggunaan (Usability): Antarmuka website harus intuitif dan mudah dipahami oleh pengguna awam.

d) Performa**:** Proses penerbitan dan verifikasi harus berjalan dengan cepat dan efisien.

3.1.2. Analisis Konseptual

Analisis ini dilakukan untuk mendapatkan landasan teoritis yang kuat dan memahami konsep-konsep kunci yang relevan dengan penelitian. Literatur yang ditinjau mencakup:

1) Teknologi *Blockchain*: Mempelajari cara kerja dasar *blockchain*, struktur data, konsep *distributed ledger technology* (DLT), dan keunggulan utamanya, yaitu desentralisasi, transparansi, dan imutabilitas.
2) *Smart Contract*: Memahami peran *smart contract* sebagai kode yang berjalan di atas *blockchain* untuk mengotomatisasi proses penerbitan dan verifikasi tanpa perantara.
3) Penerapan *Blockchain* pada Sertifikat Digital: Menganalisis penelitian-penelitian terdahulu yang telah mengimplementasikan *blockchain* untuk masalah serupa, seperti sertifikat akademik atau dokumen legal, untuk mengidentifikasi praktik terbaik dan tantangan yang mungkin dihadapi.
4) Pengembangan *Website*: Mempelajari teknologi dan kerangka kerja (framework) yang digunakan untuk membangun aplikasi web modern, termasuk *frontend* dan *backend*.

3.1.3. Analisis Teknologi

Pada tahap ini, dilakukan pemilihan teknologi yang paling sesuai untuk mendukung pengembangan sistem. Pemilihan ini didasarkan pada keandalan, fleksibilitas, dan ketersediaan sumber daya.

1) Teknologi *Blockchain*:
    a) Ethereum**:** Dipilih karena memiliki dukungan *smart contract* yang kuat (menggunakan bahasa Solidity) dan komunitas pengembang yang besar, yang memudahkan proses implementasi. Jaringan Ethereum Testnet (seperti Sepolia) akan digunakan untuk

pengembangan dan pengujian sistem tanpa biaya, sebelum berpotensi diimplementasikan pada *mainnet*.

2) Pengembangan *Website*:
   a) *Frontend*: React.js akan digunakan untuk membangun antarmuka pengguna yang dinamis dan responsif, memastikan pengalaman pengguna yang baik.
   b) *Backend*: Node.js akan dipilih sebagai *backend* untuk mengelola logika server, berinteraksi dengan database, dan berkomunikasi dengan *smart contract* di *blockchain*.
3) Alat dan Pustaka:
   a) Web3.js atau Ethers.js: Pustaka ini akan digunakan untuk menghubungkan aplikasi web dengan jaringan *blockchain* Ethereum, memungkinkan interaksi langsung dengan *smart contract*.
   b) IPFS (InterPlanetary File System): Teknologi ini bisa dipertimbangkan untuk menyimpan metadata sertifikat secara terdesentralisasi, bukan hanya hash-nya

3.2. Desain

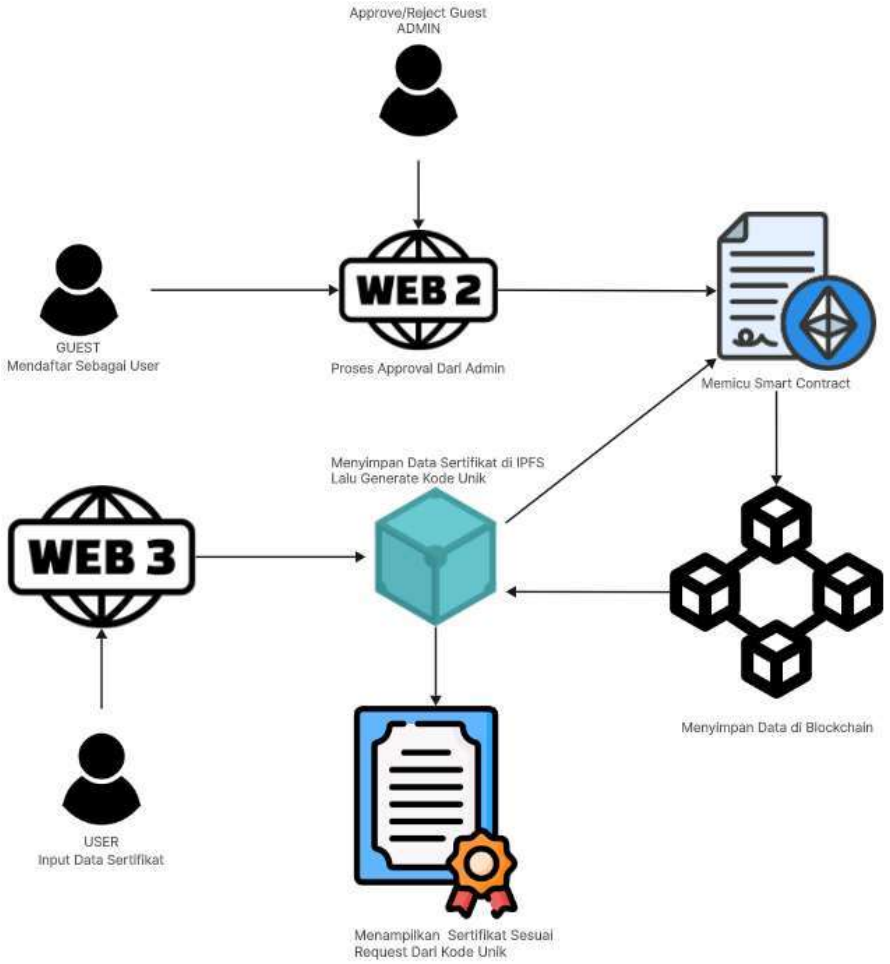
Tahap desain merupakan implementasi dari hasil analisis kebutuhan. Pada tahap ini, peneliti merancang cetak biru (blueprint) dari sistem yang akan dibangun. Desain ini mencakup struktur keseluruhan sistem, tampilan antarmuka pengguna, dan alur kerja yang akan diterapkan. Tahap ini sangat penting untuk memastikan bahwa pengembangan sistem berjalan secara terstruktur dan sesuai dengan tujuan penelitian. Berbeda dengan pendekatan konvensional, sistem ini tidak akan menggunakan database terpusat, melainkan mengandalkan sepenuhnya pada *smart contract* di *blockchain* untuk penyimpanan dan manajemen data sertifikat. Desain sistem akan dibagi menjadi beberapa sub-bagian, yaitu:

3.2.1. Desain Arsitektur Sistem

Arsitektur sistem ini menggambarkan bagaimana setiap komponen utama berinteraksi satu sama lain. Arsitektur yang digunakan adalah arsitektur terdesentralisasi penuh yang mengandalkan *smart contract* sebagai satu-satunya sumber data.

1) Aplikasi Web (Frontend): Dibangun menggunakan React.js, berfungsi sebagai antarmuka bagi pengguna dan administrator. Aplikasi ini akan terhubung langsung ke *blockchain* melalui Web3.js atau Ethers.js tanpa perantara *server* tradisional.
2) Jaringan *Blockchain*: Jaringan Ethereum Testnet akan digunakan sebagai *distributed ledger* yang tidak dapat diubah (immutable). Di sini, *smart contract* berfungsi ganda sebagai *backend* dan database, menyimpan semua data sertifikat secara permanen dan transparan.
3) *Smart Contract*: Berisi logika bisnis yang kompleks untuk menerbitkan, menyimpan, dan memverifikasi sertifikat. Semua data yang diperlukan (misalnya, hash sertifikat, nama penerima, acara, dll.) akan disimpan dalam *state smart contract* itu sendiri.

Dengan arsitektur ini, semua data sertifikat akan benar-benar terdesentralisasi, menghilangkan titik kegagalan tunggal (*single point of failure*) yang ada pada sistem dengan database terpusat.

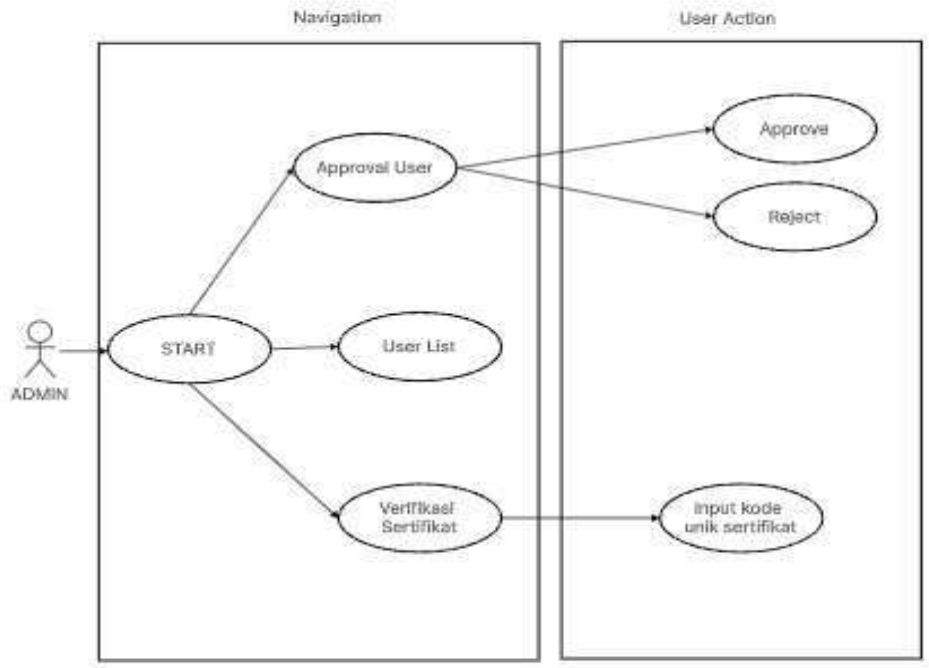


Gambar 3. 1 Diagram Arsitektur Aplikasi Web

Gambar 3.1 ini akan menunjukan gambar diagram Arsitektur web yang menggambarkan bagaimana keseluruhan aplikasi sistem aplikasi berbasis web yang dikembangkan.

3.2.2. Use Case Diagram

Use Case Diagram ini akan menunjukan gambar diagram kasus penggunaan web yang menggambarkan bagaimana interaksi antara pengguna dan system aplikasi berbasis web yang dikembangkan.



Gambar 3. 2 Use Case penggunaan web dari pandangan admin

Pada Gambar 3.2 Ketika admin berhasil login ke sistem, admin akan diarahkan ke halaman Approval User. Pada halaman ini, admin dapat melihat daftar pengguna yang baru mendaftar dan belum mendapatkan hak akses penuh.

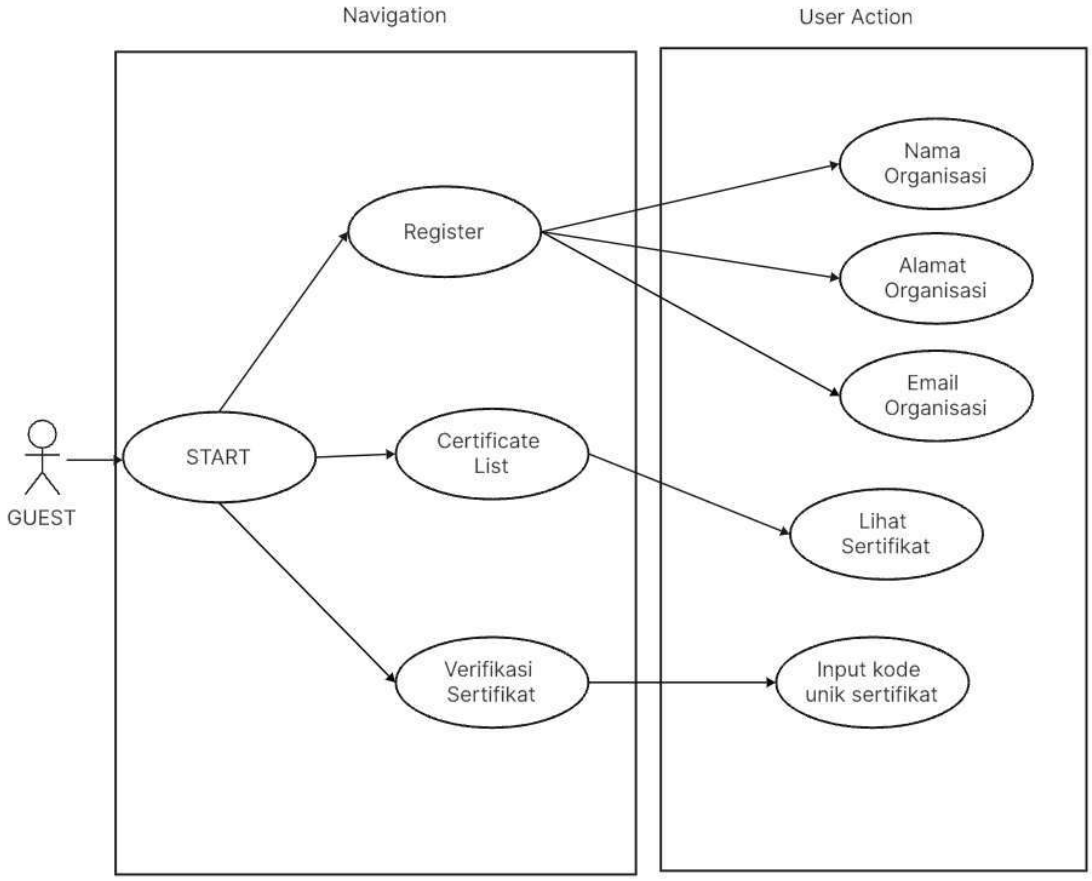
Admin memiliki pilihan untuk menyetujui (approve) atau menolak (reject) permintaan pendaftaran tersebut.

Jika permintaan approve diberikan, sistem akan memicu *smart contract* dan status pengguna akan tercatat di *blockchain* sehingga pengguna dapat mengakses fitur pembuatan sertifikat digital sesuai hak akses yang diberikan. Sebaliknya, jika admin memilih reject, data pengguna tetap ada namun statusnya ditandai sebagai ditolak, sehingga tidak dapat menggunakan fitur pembuatan sertifikat.

Selain itu, admin juga dapat mengakses halaman User List yang menampilkan seluruh data pengguna yang terdaftar dan memiliki status aktif. Pada halaman ini, admin dapat melakukan tindakan revoke terhadap pengguna tertentu. Fitur revoke digunakan untuk mencabut hak akses pengguna yang melanggar ketentuan atau tidak lagi berhak menggunakan sistem. Setelah proses revoke dilakukan, pengguna tidak dapat lagi membuat atau mengelola sertifikat digital di dalam sistem.

Admin juga memiliki akses ke halaman Verify Certificate, yang memungkinkan admin untuk melakukan verifikasi sertifikat secara manual. Pada halaman ini, admin dapat memasukkan kode unik yang terdapat pada sertifikat digital. Sistem akan mencari data sertifikat berdasarkan kode unik tersebut di *blockchain* dan/atau database. Jika data ditemukan, sistem akan menampilkan informasi detail sertifikat seperti nama peserta, nama kegiatan, tanggal kegiatan, dan status keaslian sertifikat. Jika data tidak ditemukan, sistem akan memberikan notifikasi bahwa sertifikat tersebut tidak valid atau tidak terdaftar.

Dengan adanya fitur approval, reject, dan revoke ini, admin memiliki kendali penuh terhadap siapa saja yang berhak mengakses sistem, sehingga keamanan dan validitas proses penerbitan sertifikat digital dapat lebih terjamin.



Gambar 3. 3 Use Case penggunaan web dari pandangan guest

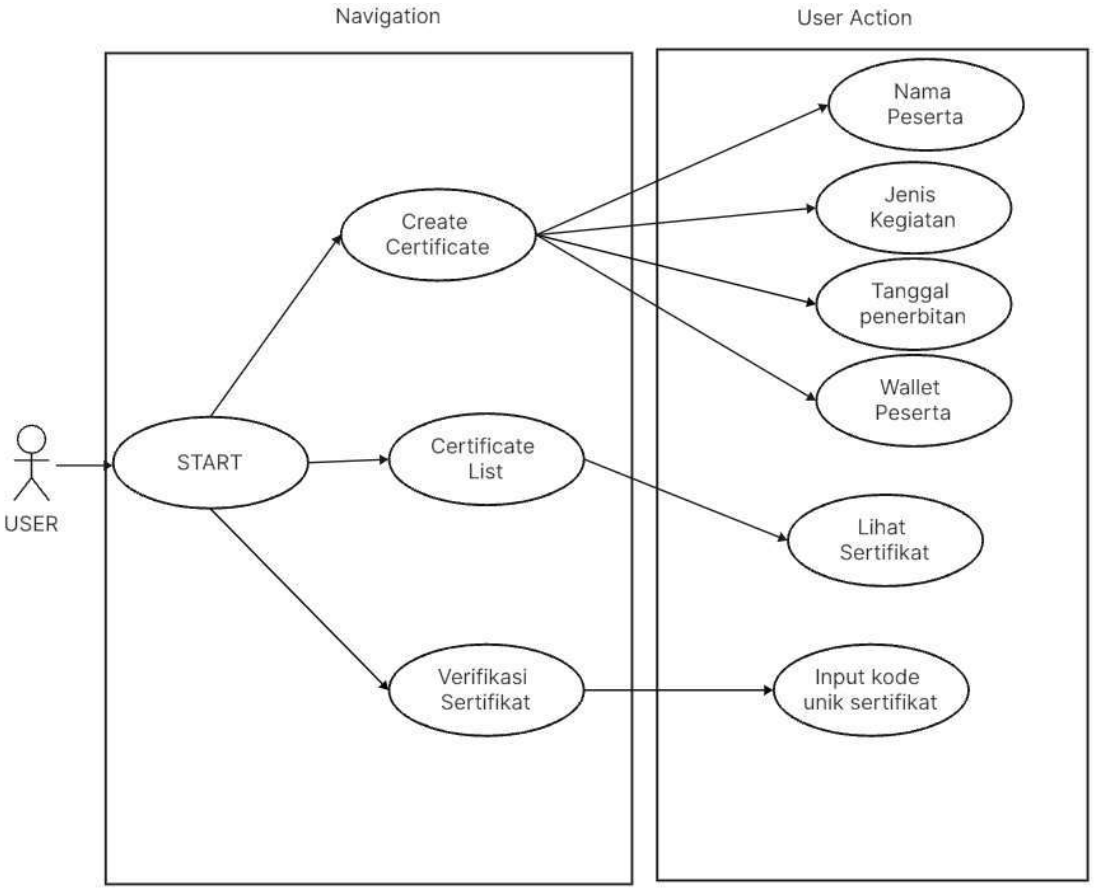
Pada Gambar 3.3 Ketika guest mengakses halaman utama sistem, guest dapat memilih menu Register untuk melakukan pendaftaran sebagai pengguna. Pada halaman pendaftaran, guest diminta untuk mengisi data berupa nama organisasi, alamat organisasi, email organisasi, serta melakukan pembayaran sebesar 0,5 ETH sebagai biaya registrasi. Setelah semua data diisi dan pembayaran berhasil, sistem akan menyimpan data pendaftaran guest dan menandainya sebagai akun yang menunggu persetujuan admin.

Setelah akun guest disetujui oleh admin, guest dapat mengakses halaman List Sertifikat. Pada halaman ini, sistem akan memeriksa wallet yang

terhubung dengan akun guest untuk mencari sertifikat NFT yang dimiliki. Jika sertifikat ditemukan, daftar sertifikat tersebut akan ditampilkan lengkap dengan informasi kegiatan, tanggal, dan statusnya. Jika guest belum memiliki sertifikat di wallet, maka halaman akan kosong atau menampilkan notifikasi bahwa tidak ada sertifikat yang dimiliki.

Selain itu, guest juga dapat mengakses halaman Verify Certificate untuk melakukan pengecekan keaslian sertifikat. Pada halaman ini, guest dapat memasukkan kode unik yang tercantum pada sertifikat digital. Sistem akan memverifikasi kode unik tersebut di *blockchain* dan/atau database. Jika valid, sistem akan menampilkan informasi detail sertifikat seperti nama peserta, nama kegiatan, tanggal kegiatan, dan status validitas. Jika tidak valid, sistem akan memberikan informasi bahwa sertifikat tersebut tidak terdaftar atau palsu.

Dengan adanya fitur registrasi, akses daftar sertifikat, dan verifikasi sertifikat, guest dapat secara mandiri mendaftar ke sistem, melihat sertifikat yang mereka miliki, serta memverifikasi keaslian sertifikat tanpa perlu bantuan admin secara langsung.



Gambar 3. 4 Use Case penggunaan web dari pandangan user

Pada Gambar 3.4 Setelah berhasil login ke sistem, User/Organisasi dapat mengakses halaman Create Certificate. Pada halaman ini, User menginputkan data sertifikat berupa nama peserta, jenis kegiatan, tanggal penerbitan, dan alamat wallet peserta. Setelah semua data terisi, sistem akan secara otomatis membuat sertifikat digital dalam format PDF, menyimpannya ke IPFS, serta mencetaknya sebagai NFT di *blockchain* lalu mengirimkannya ke wallet peserta sesuai alamat yang diinputkan.

Selanjutnya, User dapat membuka halaman Certificate List, yang menampilkan seluruh sertifikat yang telah dibuat oleh User tersebut. Daftar ini

berisi informasi sertifikat seperti nama peserta, jenis kegiatan, tanggal penerbitan, kode unik, serta status sertifikat. Sertifikat-sertifikat ini ditampilkan berdasarkan data yang telah diinputkan User sebelumnya, dan dapat digunakan sebagai referensi atau untuk pengecekan riwayat penerbitan.

Selain itu, User juga dapat mengakses halaman Verify Certificate untuk melakukan pengecekan keaslian sertifikat. Pada halaman ini, User dapat memasukkan kode unik yang terdapat pada sertifikat digital. Sistem akan melakukan pencarian data berdasarkan kode unik tersebut di *blockchain* dan/atau database. Jika data ditemukan dan valid, sistem akan menampilkan informasi detail sertifikat seperti nama peserta, nama kegiatan, tanggal penerbitan, dan status validitasnya. Jika tidak ditemukan, sistem akan memberikan informasi bahwa sertifikat tersebut tidak terdaftar atau palsu.

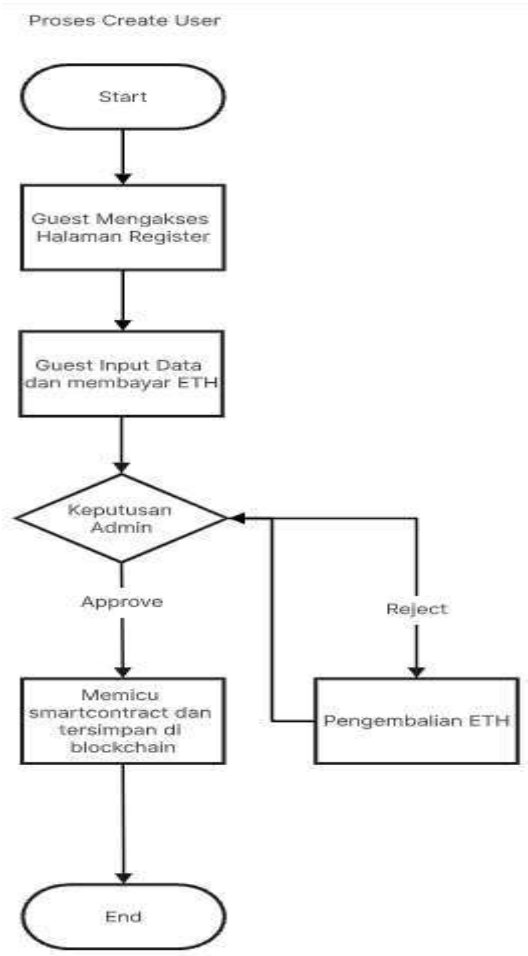
Dengan adanya fitur pembuatan sertifikat, daftar sertifikat, dan verifikasi sertifikat, User/Organisasi dapat secara mandiri menerbitkan, mengelola, dan memverifikasi sertifikat digital secara aman dan terdesentralisasi.

3.2.3. Flowchart Sistem

Bagian ini menyajikan *flowchart* sebagai representasi visual dari alur kerja sistem yang telah diimplementasikan. *Flowchart* ini dirancang untuk memetakan setiap proses fungsional, seperti alur pendaftaran, penerbitan sertifikat berbasis NFT, dan verifikasi data, guna memberikan pemahaman yang komprehensif mengenai interaksi antar komponen sistem.

1.Proses Pendaftaran

Proses pendaftaran pengguna Akan dijelaskan dari input data, pembayaran ETH, hingga keputusan persetujuan atau penolakan oleh admin dalam sistem CertiChain dapat divisualisasikan melalui *flowchart* berikut :

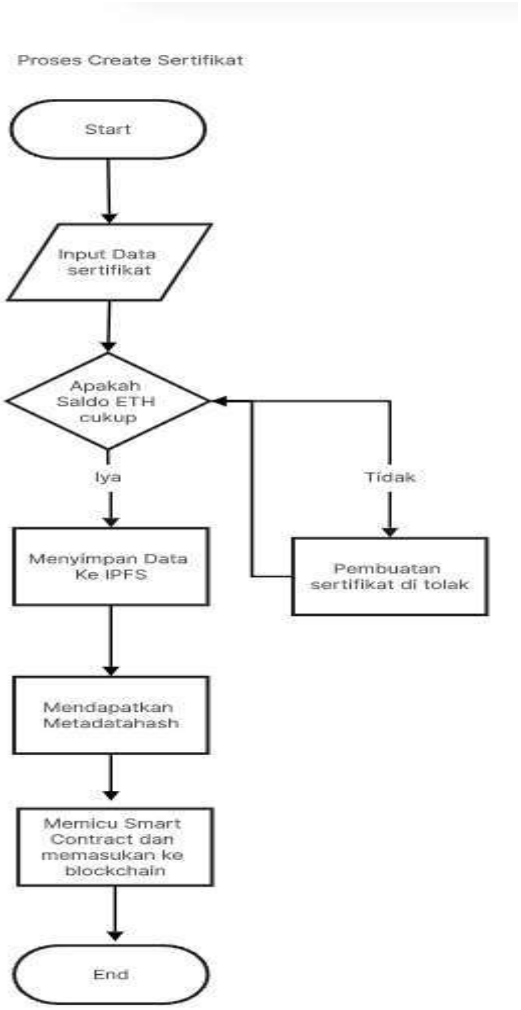


Gambar 3. 5 Flowchart Proses Pendaftaran User

Pada Gambar 3.5 Proses pendaftaran dimulai ketika seorang pengguna baru (Guest) mengakses halaman registrasi. Pada tahap ini, pengguna diwajibkan untuk mengisi data pendaftaran dan melakukan pembayaran sebesar 0.5 ETH ke alamat dompet admin. Setelah transaksi pembayaran dan pengisian data selesai, permohonan pendaftaran akan masuk dalam antrean untuk ditinjau oleh admin.

2.Proses Penerbitan Sertifikat

Proses penerbitan sertifikat dalam sistem CertiChain dapat divisualisasikan melalui *flowchart* yang menjelaskan alur dari input data sertifikat hingga tercatat di *blockchain* seperti berikut :



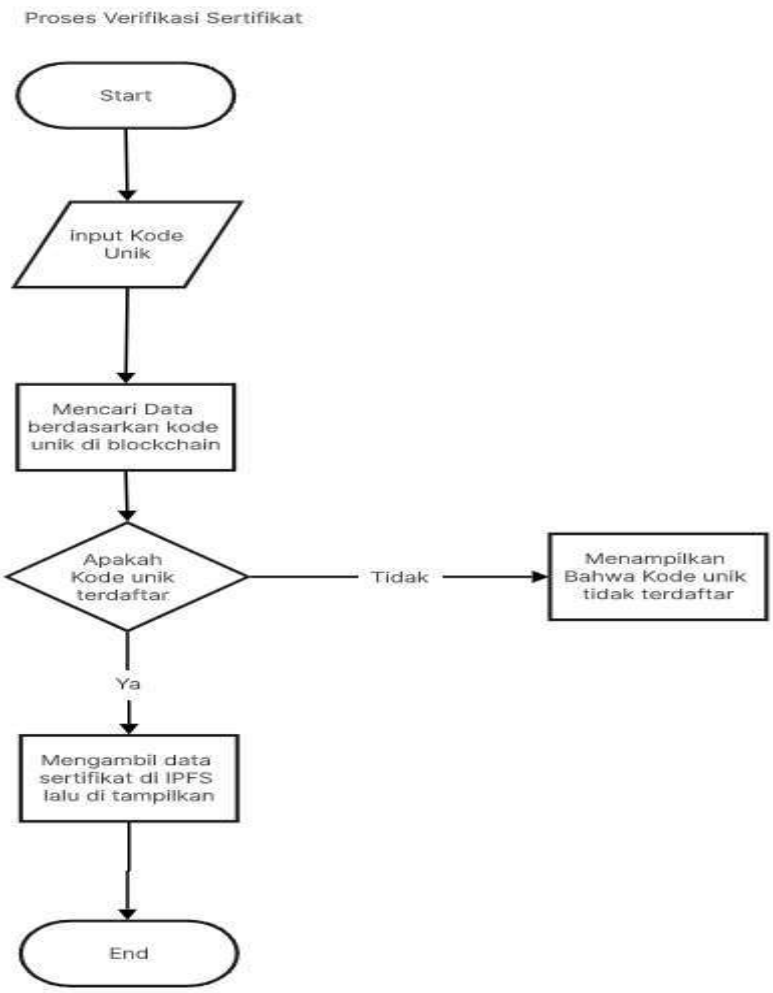
Gambar 3. 6 Flowchart Proses Penerbitan Sertifikat

Pada Gambar 3.6 Proses penerbitan sertifikat dimulai dengan input data sertifikat oleh pengguna yang telah disetujui. Setelah data dimasukkan, sistem akan memeriksa apakah saldo ETH di dompet pengguna mencukupi untuk

biaya transaksi *minting*. Jika saldo tidak mencukupi, proses akan ditolak dan pengguna tidak dapat melanjutkan. Namun, jika saldo mencukupi, data sertifikat akan disimpan ke IPFS dan sistem akan mendapatkan *hash* metadata dari IPFS tersebut. Selanjutnya, sistem akan memicu *smart contract* untuk membuat sertifikat baru sebagai NFT, dengan *hash* metadata yang didapat dari IPFS. Setelah transaksi disetujui, sertifikat tersebut akan tercatat secara permanen di *blockchain*, dan proses penerbitan pun selesai.

3.Proses Verifikasi Sertifikat

Proses verifikasi sertifikat dalam sistem CertiChain dapat divisualisasikan melalui *flowchart* berikut, yang menjelaskan alur dari input kode unik hingga tampilnya data verifikasi:



Gambar 3. 7 Flowchart Proses Verifikasi Sertifikat

Pada Gambar 3. 7 Proses verifikasi dimulai saat pengguna, siapa pun itu, memasukkan kode unik sertifikat ke dalam kolom yang tersedia. Sistem kemudian akan menggunakan kode unik tersebut untuk mencari data yang relevan di *blockchain*.

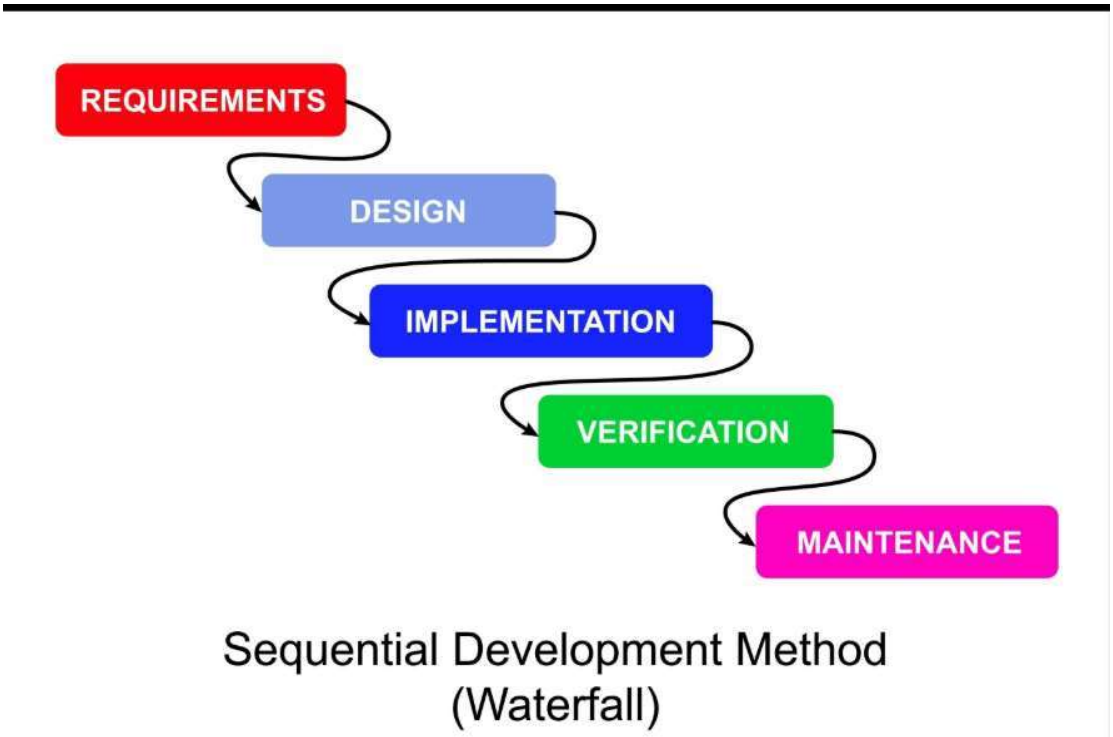
Pada tahap ini, sistem akan memeriksa apakah kode unik yang dimasukkan terdaftar di *blockchain*. Jika kode unik tidak ditemukan, proses akan berakhir dengan menampilkan pesan bahwa kode unik tidak terdaftar. Sebaliknya, jika kode unik ditemukan, sistem akan melanjutkan dengan mengambil data sertifikat yang

tersimpan di IPFS, kemudian menampilkannya di layar pengguna, termasuk detail seperti nama pemilik, jenis sertifikat, dan tanggal penerbitan. Dengan demikian, proses verifikasi selesai, dan keaslian sertifikat dapat terbukti secara transparan.

3.3. Pengembangan

Tahap pengembangan merupakan proses implementasi rancangan sistem CertiChain ke dalam bentuk aplikasi web yang fungsional. Sistem ini dikembangkan dengan pendekatan Full-Stack DApp (Decentralized Application), yang mengintegrasikan frontend dengan backend berbasis *smart contract* di jaringan *blockchain*. Fokus pengembangan diarahkan pada tiga komponen utama: Smart Contract/*Blockchain*, Backend (Integrasi Web3 & IPFS), dan Frontend (Antarmuka Pengguna).

3.3.1. Pengembangan Waterfall



Gambar 3. 8 Tahapan Pengembangan Waterfall

Sistem CertiChain dapat diuraikan menggunakan metode pengembangan waterfall yang terdiri dari beberapa tahapan utama, mulai dari analisis kebutuhan hingga pemeliharaan.

1. Tahap Analisis dan Perencanaan

Tahap ini melibatkan pengumpulan dan pendokumentasian semua persyaratan untuk sistem CertiChain. Berdasarkan deskripsi, beberapa hal yang dapat diidentifikasi di sini adalah:

a) Identifikasi Fungsi Utama: Kebutuhan untuk mengelola sertifikat digital sebagai NFT, pendaftaran dan otorisasi pengguna (Admin dan User), pembuatan sertifikat, dan verifikasi sertifikat.
b) Identifikasi Teknologi: Menentukan teknologi yang akan digunakan, seperti Solidity (untuk smart contract), React (untuk frontend), ethers.js (untuk integrasi blockchain), IPFS (untuk penyimpanan terdesentralisasi), dan alat pengembangan seperti Truffle dan Ganache.
c) Identifikasi Entitas: Menentukan peran pengguna: Admin, User/Organisasi, dan Guest.
d) Perancangan Alur Kerja: Merencanakan alur pendaftaran, pembuatan sertifikat, dan proses verifikasi.

2. Tahap Desain

Pada tahap ini, arsitektur sistem dirancang secara rinci. Ini termasuk desain smart contract, backend, dan frontend.

a) Desain Smart Contract: Merancang kontrak pintar ERC-721 yang memiliki fungsi khusus untuk minting NFT dan menyimpan metadata. Desain juga mencakup fungsi untuk menyetujui/menolak pendaftaran pengguna.
b) Desain Backend: Merancang modul integrasi Web3 menggunakan ethers.js, manajemen identitas dengan Web3Modal, dan integrasi IPFS untuk

penyimpanan file. Ini juga mencakup desain mekanisme penyimpanan sementara data pendaftaran (misalnya, di localStorage).

c) Desain Frontend: Merancang antarmuka pengguna (UI) menggunakan React dan Material UI. Ini mencakup desain tiga dashboard utama (Admin, User/Organisasi, dan Guest) serta modul-modul fungsional seperti pendaftaran, pembuatan sertifikat, dan verifikasi.

3. Tahap Implementasi (Pengembangan)

Tahap ini adalah di mana kode program ditulis. Ini mencakup pengembangan smart contract, backend, dan frontend secara terpisah namun terkoordinasi.

a) Pengembangan Smart Contract: Menulis kode Solidity ^0.8.20 untuk kontrak ERC-721 dengan semua fungsi yang dirancang.

b) Pengembangan Backend: Menulis kode untuk mengintegrasikan ethers.js, Web3Modal, dan ipfs-http-client.

c) Pengembangan Frontend: Mengembangkan UI dengan React 18 dan Material UI, mengimplementasikan modul pendaftaran, pembuatan sertifikat menggunakan jsPDF dan uuid, serta modul verifikasi.

4. Tahap Pengujian

Setelah implementasi selesai, setiap komponen diuji untuk memastikan fungsinya berjalan dengan benar sesuai dengan desain.

a) Pengujian Smart Contract: Menggunakan Truffle dan lingkungan Ganache untuk menguji fungsionalitas seperti proses pendaftaran, minting NFT, dan penyimpanan metadata di blockchain.

b) Pengujian Integrasi: Menguji interaksi antara frontend, backend, dan smart contract. Memastikan transaksi berjalan lancar, dan data dapat dibaca serta ditulis ke blockchain.

c) Pengujian Fungsionalitas Frontend: Menguji semua modul (pendaftaran, pembuatan sertifikat, dan verifikasi) untuk memastikan antarmuka pengguna responsif dan alur kerja berjalan sesuai harapan.

d) Pengujian Penanganan Kesalahan: Memastikan sistem memberikan notifikasi yang jelas jika terjadi kegagalan (misalnya, koneksi terputus atau transaksi gagal).

5. Tahap Pemasangan dan Pemeliharaan

Meskipun deskripsi Anda berfokus pada lingkungan pengembangan lokal, tahap ini secara konseptual adalah tentang penyebaran sistem ke lingkungan produksi dan pemeliharaan jangka panjang.

a) Penyebaran (Deployment): Menyebarkan smart contract ke jaringan blockchain publik (misalnya, Ethereum Mainnet atau testnet). Menerbitkan aplikasi web (frontend dan backend) ke server.

b) Pemeliharaan: Memantau sistem, memperbaiki *bug* yang muncul, dan melakukan pembaruan atau peningkatan sesuai kebutuhan di masa depan. Misalnya, memperbarui versi pustaka yang digunakan atau menambahkan fitur baru.

3.4. Evaluasi

3.4.1. Evaluasi Sistem

Evaluasi sistem dilakukan untuk memastikan bahwa seluruh fitur yang telah diimplementasikan berjalan sesuai kebutuhan dan menghasilkan keluaran yang diharapkan. Pengujian dilakukan dengan beberapa metode

1) Pengujian Black Box

Pengujian ini dilakukan untuk memverifikasi fungsionalitas sistem dari sudut pandang pengguna tanpa melihat kode internal. Tujuannya

adalah memastikan setiap alur utama berjalan sesuai kebutuhan dan data tersimpan dengan benar di *blockchain*.

a) Alur Pendaftaran dan Persetujuan diuji untuk memastikan Guest dapat mendaftar, serta Admin dapat menyetujui atau menolak permintaan sehingga status pengguna berubah di *blockchain*.
b) Alur Penerbitan Sertifikat diuji agar User/Organisasi dapat membuat sertifikat, mengunggahnya ke IPFS, serta mencetak NFT engan metadata yang benar (termasuk CID dan kode unik).
c) Alur Verifikasi Sertifikat diuji untuk memastikan input kode unik valid dapat menampilkan data sertifikat dari *blockchain* secara akurat, sementara input yang tidak valid memunculkan pesan peringatan.

Tabel 3. 2 Sampel Skema Pengujian Blackbox Pada sistem

No	Menu	Fungsi yang Diuji	Deskripsi	Hasil yang Diharapkan
1	Register	Pendaftaran User Baru	User mengisi form registrasi dan mengirim 0.5 ETH ke admin	Transaksi 0.5 ETH berhasil, status pending tersimpan, pesan sukses muncul
2	Register	Validasi Form Kosong	User mencoba submit form tanpa mengisi semua field	Pesan error "Semua field harus diisi" muncul, tidak ada transaksi

No	Menu	Fungsi yang Diuji	Deskripsi	Hasil yang Diharapkan
3	Approve Users	Approval User Pending	Admin approve user yang status pending	User berhasil ditambahkan ke *smart contract*, status jadi approved, pesan sukses
4	Approve Users	Reject User Pending	Admin reject user yang status pending	Refund 0.5 ETH otomatis ke user, status jadi rejected, pesan sukses
5	Create Certificate	Pembuatan Sertifikat	User approved membuat sertifikat baru	PDF berhasil dibuat, upload IPFS sukses, mint NFT berhasil, pesan sukses
6	Create Certificate	Buat Sertifikat Tanpa Registrasi	User non-approved mencoba buat sertifikat	Gagal mint, pesan error kontrak muncul
7	Verify Certificate	Verifikasi Kode Valid	Input kode unik yang sudah ada	Data sertifikat tampil lengkap, link IPFS bisa dibuka
8	Navigation	Menu Admin	Connect wallet	Menu Admin, List User,

No	Menu	Fungsi yang Diuji	Deskripsi	Hasil yang Diharapkan
			sebagai admin	Approve Users tampil
9	System	Ganti Network	Ganti network di wallet	Aplikasi reload otomatis, koneksi diinisialisasi ulang
10	System	Ganti Akun	Ganti akun di wallet	Menu berubah sesuai role akun baru
11	Data	Konsistensi Data	Mint sertifikat lalu verifikasi	Data verifikasi cocok dengan input mint
12	Error	Email Invalid	Input email format salah	Validasi email gagal, form tidak submit

2) Pengujian Integrasi

a) Pengujian ini berfokus pada interaksi antara berbagai komponen sistem, terutama antara frontend, dompet kripto, dan *smart contract* di *blockchain*.

b) Interaksi Frontend-Web3: Dipastikan bahwa Web3Modal dapat berhasil terhubung dengan dompet pengguna (seperti MetaMask) dan bahwa ethers.js dapat mengirim transaksi ke *smart contract* dengan benar.

c) Interaksi *Blockchain*-IPFS: Diuji bahwa CID yang didapat dari IPFS benar-benar disimpan di dalam *blockchain* sebagai bagian dari metadata NFT dan dapat diambil kembali saat proses verifikasi.

d) Ketahanan Jaringan: Meskipun menggunakan jaringan lokal, pengujian ini memastikan bahwa setiap interaksi dengan *blockchain* dapat ditangani dengan baik, termasuk *error handling* jika transaksi gagal atau jaringan terputus.

Tabel 3. 3 Sampel Skema Pengujian Integrasi

NO	Modul yang Diuji	Deskripsi Skenario Pengujian
1	Frontend & Web3	Koneksi Wallet ke Frontend
2	Frontend & Web3	Switch Network
3	Frontend & Smart Contract	Pendaftaran User
4	Admin & Smart Contract	Approval User
5	Frontend, Smart Contract, & IPFS	Pembuatan Sertifikat Lengkap
6	Frontend & Smart Contract	Verifikasi Sertifikat
7	Frontend, IPFS, & Smart Contract	Akses File Sertifikat
8	Frontend & Local Storage	Sesi Pendaftaran
9	Admin & Smart Contract	Reject User & Refund

3) Pengujian Korelasi Pearson

Pengujian ini dilakukan untuk mengevaluasi apakah data yang diinput ke sistem memiliki korelasi dengan hasil hash yang disimpan di *blockchain* melalui algoritma *Keccak-256*. Tujuannya adalah untuk membuktikan

bahwa hasil hash tidak menunjukkan hubungan linear dengan input, sehingga data lebih aman dan tidak dapat ditebak.

a) Nilai korelasi Pearson dihitung antara representasi numerik dari data input dengan hasil hash *Keccak-256* di Ganache.
b) Hasil pengujian menunjukkan bahwa nilai korelasi mendekati nol, yang menandakan hash bersifat acak, tidak memiliki pola linear, dan aman digunakan dalam sistem *blockchain*.

Tabel 3. 4 Sampel Skema Pengujian Korelasi Pearson

No	Inputan	Hash Keccak 256
1	LSP DEF 0xCD77Cd044B9d70e8Ae4D5239 e523a5bCADf4E941 Jakarta LSPDEF@gmail.com BNSP-LSP-157-ID	0xf08cd4840bf5d51b2118c 8370506c9d0f05c90dd8032 c445f89d184cebd09bd3
2	Kemendikbud 0x5Fef27795C774764cEAbb22C6 FA62641E84e4fEF Jakarta Kemendikbud@gmail.com	0x46366c5e72b3f9b89e424 7cff6c2fa96d7d5369027ff4c 5f6d2308cb99f93daf
3	LSP ABC 0x77948c4a176544279CCdCaA78 F032956657b3FE5 Jakarta LSPDEF@gmail.com BNSP-LSP-157-ID	0x547b2b24bc6784d2ea966 08bdc3b752315934c4c64f8 2477a47bd87437e131be
4	LSP GHI 0x812E34d92500571c6C3698C8F C5d35D3D995cddB Bandung LSPGHI@gmail.com BNSP-LSP-333-ID	0xc9647bb4142bc8ebc04c0 ee1eb43328b4a3f197b9519 14e6329bfcae19a2869b

5	LSP JKL 0x507D51c428B139cC3CaDb5691 0C68ad0524c9192 Lampung LSPJKL@gmail.com BNSP-LSP-445-ID	 0x41836ff263eb9c12ac44d6 42298319585371fedd2ff824 1201ec618e63dc8001