Implementasi Blockchain pada Sistem Presensi di Platform Android

Violia Ruana Nur'aini Sagita[1], I Made Suartana[2]

[1,2] S1 Teknik Informatika, Universitas Negeri Surabaya
[1]violiaruana.21004@mhs.unesa.ac.id
[2]madesuartana@unesa.ac.id

***Abstrak*— Sistem presensi konvensional yang masih banyak digunakan saat ini memiliki berbagai kelemahan, seperti potensi manipulasi data dan ketergantungan pada server terpusat. Untuk mengatasi permasalahan tersebut, penelitian ini bertujuan mengembangkan sistem presensi berbasis mobile yang terintegrasi dengan teknologi blockchain guna meningkatkan keamanan, validitas, dan integritas data kehadiran. Metode penelitian yang digunakan adalah *Agile Blockchain Dapp Engineering (ABCDE)*, dimulai dari identifikasi kebutuhan, desain sistem, pengembangan aplikasi, hingga tahap pengujian dan *deployment*. Sistem dikembangkan menggunakan Flutter untuk platform Android, Elysia.js dan Bun untuk *backend*, serta integrasi *database* MySQL dan SQLite. Proses pencatatan presensi dilakukan melalui blok data yang terhubung secara kriptografis dan didistribusikan menggunakan jaringan *peer-to-peer (P2P)*. Hasil pengujian menunjukkan bahwa data presensi yang tercatat di *database* server dan mobile bersifat konsisten, serta valid berdasarkan parameter *hash* dan *nonce*. Selain itu, uji beban menggunakan Grafana k6 menunjukkan bahwa aplikasi mampu menangani 20 *virtual users* dengan rata-rata latensi 239,8 ms, yang masih dalam batas optimal. Berdasarkan hasil tersebut, dapat disimpulkan bahwa integrasi blockchain pada sistem presensi mobile berhasil meningkatkan keamanan dan reliabilitas data.**

***Kata Kunci* — Sistem presensi, blockchain, Agile Blockchain DappEngineering (ABCDE), validitas data, beban aplikasi, functional testing**

I. PENDAHULUAN

Dalam beragam dimensi kehidupan baik dalam lembaga formal dan lembaga non formal, sistem presensi memainkan peran penting. Pencatatan kehadiran secara konvensional dilakukan secara manual, dengan mencantumkan nama lengkap dan tanda tangan sebagai bukti kehadiran [1]. Dalam penerapannya, metode konvensional ini cukup sederhana dan mudah diterapkan. Akan tetapi, masih memiliki berbagai kelemahan seperti hilangnya buku presensi, potensi kesalahan dalam pencatatan, atau bahkan manipulasi data yang dilakukan oleh pihak yang tidak bertanggung jawab. Selain itu, proses manual sering kali memakan waktu dan tenaga, terutama ketika data kehadiran harus direkapitulasi secara berkala untuk keperluan laporan.

Sejalan dengan kemajuan teknologi, kini sistem presensi telah digitalisasi dengan memanfaatkan sistem basis data atau *database* sebagai penyimpanan data. Penggunaan sistem presensi berbasis elektronik atau digital ini menawarkan berbagai keuntungan karena sistem ini dapat menyajikan informasi kehadiran secara *real-time* dan langsung terintegrasi ke dalam *database* yang dikelola secara otomatis. Sehingga, data kehadiran dapat direkam dan diakses kapan saja tanpa melalui proses manual yang rumit. Sistem ini juga mendukung pengelolaan data kehadiran secara massal dengan lebih tepat, efisien, dan terorganisir, sehingga laporan kehadiran dapat disajikan dengan cepat dan tanpa risiko kehilangan data.

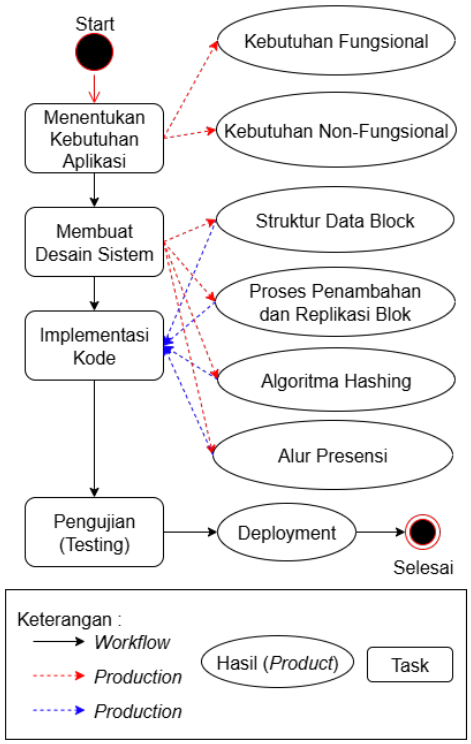
Walaupun penerapan sistem presensi berbasis elektronik memberikan efisiensi lebih dibandingkan cara manual, sistem ini masih memiliki kelemahan. Menurut penelitian sebelumnya yang berjudul "Implementasi Flutter Pada Aplikasi Presensi Karyawan Berbasis Mobile", sistem presensi yang dibuat oleh peneliti sebelumnya hanya bergantung pada satu server pusat [2]. Dengan demikian, terdapat peluang bagi pengguna untuk mengatur datanya. Oleh sebab itu, penulis berencana untuk mengembangkan lebih jauh dengan memanfaatkan teknologi terbaru, yaitu blockchain.

Berdasarkan pendapat sejumlah pakar dari jurnal penelitian sebelumnya, mereka setuju bahwa blockchain merupakan solusi yang sangat efektif untuk keamanan dan keabsahan data yang dapat diandalkan, karena teknologi ini mencatat setiap transaksi atau perubahan data dengan transparansi dan keamanan [3]. Semua data disimpan dalam blok yang saling terhubung menggunakan mekanisme *hashing* kriptografi. Teknologi ini tidak hanya memastikan bahwa data yang disimpan tidak dapat diubah atau dimanipulasi, tetapi juga terdistribusi di berbagai titik dalam jaringan (*decentralized*). Dengan kata lain, setiap perubahan atau transaksi harus melewati proses verifikasi oleh jaringan, sehingga mengurangi peluang manipulasi data oleh pihak tertentu.

Dalam penelitian ini, penulis merancang dan mengembangkan sebuah sistem presensi karyawan berbasis mobile yang diintegrasikan dengan teknologi blockchain. Pengembangan sistem ini dilakukan dengan memanfaatkan *framework* Flutter pada sisi antarmuka pengguna, serta menggunakan Elysia.js, MySQL, dan Redis sebagai sistem antrean (*queue*) untuk mendukung pengelolaan antrean data presensi secara *real-time* sebelum disimpan ke dalam *database*. Blockchain diimplementasikan guna memastikan integritas dan keamanan data kehadiran melalui pencatatan data dalam bentuk blok yang terhubung secara kriptografis. Diharapkan dari penelitian ini dapat dihasilkan sistem presensi yang tidak hanya efisien dan mudah digunakan, tetapi juga mampu mencegah manipulasi data dengan pendekatan desentralisasi, sehingga memberikan kontribusi nyata terhadap pengembangan sistem presensi digital yang aman dan terpercaya.

II. METODE PENELITIAN

Dalam penelitian ini, penulis memilih menggunakan metode *Agile BlockChain Dapp Engineering (ABCDE)*. Metode *ABCDE* adalah metode yang menggabungkan komponen blockchain dan komponen internal serta eksternal rantai, bersama-sama untuk membentuk sistem perangkat lunak desentralisasi lengkap [12]. Penggunaan metode ini dipilih karena metode ini dirancang untuk mengelola masalah khusus *BOS (Blockchain-Oriented Software)* dalam fase desain dan implementasi dApps. Penelitian ini dimulai dari menentukan kebutuhan aplikasi, membuat desain sistem, melakukan implementasi kode, pengujian (*testing*), dan *deployment*. Alur penelitian ini dijelaskan pada Gbr. 1.



Gbr. 1 Alur Rancangan Penelitian

A. Requirement

Pada tahap ini, penulis melakukan teknik elisitasi berupa analisis dokumen untuk mengumpulkan data mengenai kebutuhan fungsional dan non-fungsional dalam sistem presensi. Analisis dokumen adalah metode pengumpulan data yang melibatkan inspeksi terhadap dokumen yang sudah ada, seperti prosedur operasi, *flowchart*, dan dokumen lainnya yang menjelaskan suatu proses. Dokumen-dokumen ini memberikan konteks dan latar belakang yang diperlukan untuk menganalisis kebutuhan [2], [4], [13]. Selain itu, dokumen tersebut juga dapat dijadikan komparasi dengan sistem presensi serupa yang tidak terintegrasi dengan blockchain. Dalam dokumen-dokumen tersebut juga terdapat kelemahan, karena masih bergantung pada satu sistem terpusat. Oleh karena itu, kebutuhan aplikasi yang dikembangkan oleh penulis sekarang, yakni sistem presensi dengan integrasi blockchain dapat dijadikan solusi untuk permasalahan tersebut. Adapun kebutuhan yang diperlukan pada penelitian ini yaitu sebagai berikut:

1) Kebutuhan Fungsional

- Sistem harus menyediakan fitur untuk mendaftar akun bagi pengguna baru dengan menggunakan data identitas yang valid.
- Sistem harus menyediakan fitur untuk *login* pengguna dengan menggunakan *username* dan *password* yang telah didaftarkan.
- Sistem harus menyediakan fitur bagi pengguna yang telah terotentikasi untuk melakukan presensi masuk (*clock in*) dan presensi keluar (*clock out*) melalui perangkat Android.
- Sistem harus menyediakan halaman untuk menampilkan riwayat presensi pengguna, yang mencakup informasi waktu presensi, jenis presensi (*clock in* atau *clock out*), serta riwayat blockchain (*previous hash* dan *blockchain hash*).
- Sistem harus mampu mereplikasi data antar *node* menggunakan teknologi *peer-to-peer (p2p).*

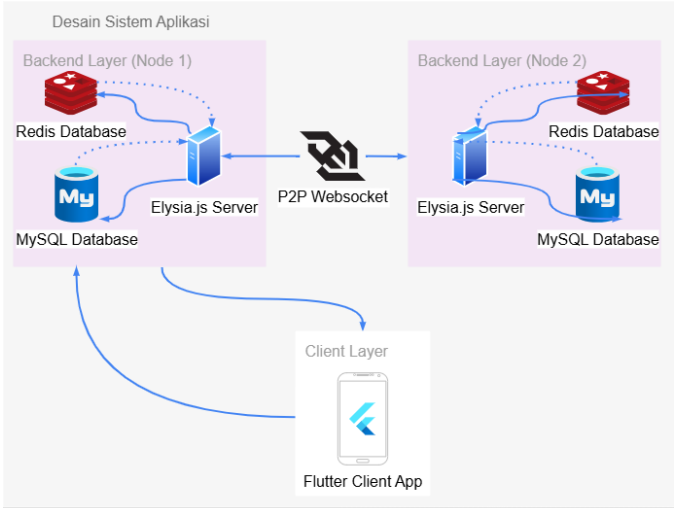
2) Kebutuhan Non-Fungsional

- Sistem harus mampu menangani peningkatan jumlah pengguna dan transaksi tanpa penurunan performa. Pengujian dilakukan menggunakan Grafana K6, dan nilai latensi maksimum tidak boleh melebihi 300 milidetik selama simulasi beban.
- Data presensi ke *database* lokal (SQLite) dan *database* server (MySQL) harus dicatat secara konsisten dan *idempotent* oleh sistem.
- Seluruh data presensi harus dihash menggunakan algoritma SHA-256. Selain itu, setiap blok harus memiliki hash sebelumnya dari hash blok sebelumnya dan tidak dapat diubah.

B. Design

Setelah analisis kebutuhan selesai, rancangan desain sistem dibuat, yang mencakup desain sistem aplikasi, implementasi blockchain, algoritma *hashing*, proses penambahan dan replikasi blok, hingga alur sistem presensi. Berikut ini merupakan rincian dari rancangan desain pada sistem presensi:

1) Desain Sistem Aplikasi

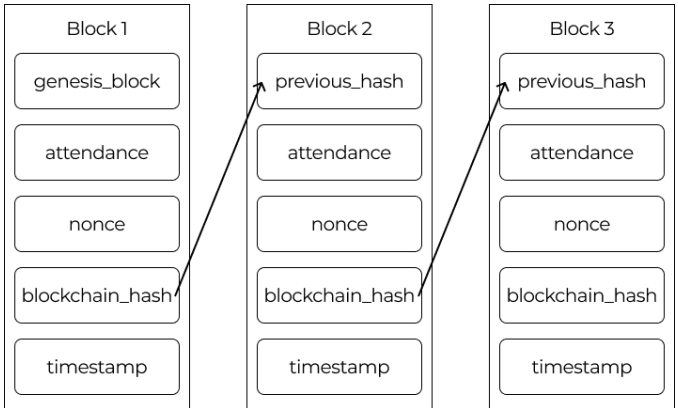


Gbr. 2 Desain Sistem Aplikasi

Pada Gbr. 2 di atas merupakan gambaran sistem presensi yang aman dan tahan terhadap manipulasi data yang menggunakan prinsip blockchain dan arsitektur terdistribusi berbasis teknologi *peer-to-peer*. *Backend* memiliki dua *node* yang terhubung ke server Elysia.js, *database* MySQL, dan *cache* Redis. Elysia.js menangani permintaan klien dan komunikasi antar komponen, sementara MySQL menyimpan data permanen, sedangkan Redis berfungsi sebagai sistem antrean (*queue*) untuk mendukung pengelolaan antrean data presensi secara *real-time* sebelum disimpan ke dalam *database*. *Node backend* berkolaborasi dengan WebSocket untuk memastikan pertukaran data presensi yang aman dan konsisten, mengadopsi konsep blockchain untuk menjaga integritas data. Aplikasi mobile berbasis Flutter menawarkan antarmuka pengguna untuk *register*, *login*, presensi, dan riwayat presensi. Kemusian, *node backend* memproses dan menyebarkan data ke *node* lainnya.

2) Struktur Data Block

Pada Gbr. 3, struktur data blok digambarkan sebagai daftar blok yang saling terhubung. Setiap blok mengandung data dan *hash* dari blok sebelumnya untuk menjamin integritas data. Berikut adalah penjelasan dari elemen-elemen pada struktur data ini:



Gbr. 3 Struktur Data Block

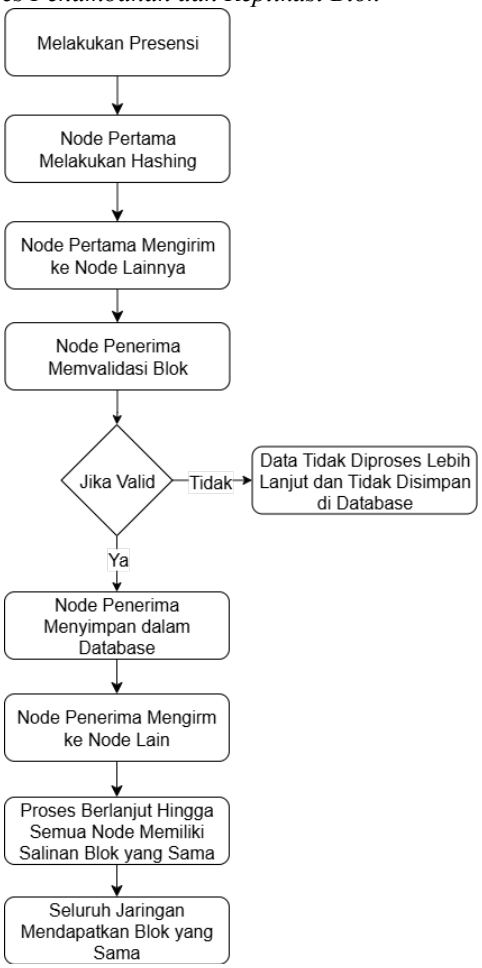
Struktur data dalam blockchain terdiri dari beberapa elemen utama yang tersusun dalam setiap blok. Blok pertama disebut *genesis block* dan merupakan titik awal dari rantai blockchain karena tidak memiliki *hash* sebelumnya. Selanjutnya, setiap blok selain *genesis block* memiliki elemen *previous hash* yang berisi nilai *hash* dari blok sebelumnya sebagai penghubung antar blok dalam membentuk rantai yang utuh. Salah satu data utama dalam setiap blok adalah *attendance*, yaitu informasi terkait catatan kehadiran pengguna. Untuk menjaga keamanan, digunakan *nonce*, yakni bilangan yang divariasikan dalam proses pencarian *hash* yang memenuhi kriteria tertentu. Hasil dari proses ini disebut *blockchain hash*, yaitu nilai unik yang dihasilkan dari kombinasi data dalam blok dan digunakan untuk memastikan integritas data. Kemudian, terdapat elemen *timestamp* yang mencatat waktu pembuatan blok, menandai kapan blok tersebut ditambahkan ke dalam jaringan blockchain.

3) Algoritma Hashing

Fungsi *hash* kriptografis bernama SHA-256 digunakan untuk memastikan setiap blok data dalam sistem blockchain tetap unik dan aman dari manipulasi. Fungsi ini memainkan peran penting dalam proses validasi data karena menghasilkan *output* berupa deretan angka biner sepanjang 256 bit yang unik untuk setiap input. Salah satu keunggulan utama SHA-256 adalah sensitivitasnya yang tinggi terhadap perubahan data sekecil apa pun, bahkan jika hanya satu bit dari input diubah. Fenomena ini dikenal sebagai efek *avalanche*, dan berperan besar dalam mendeteksi modifikasi data.

SHA-256 telah diuji dan terbukti tahan terhadap serangan collision (dua input menghasilkan *hash* yang sama) dan *preimage* (menebak input berdasarkan output hash), dengan tingkat keamanan kriptografis setara 2^128 menurut *The Science and Information Organization* [14]. SHA-256 banyak digunakan dalam sistem blockchain, seperti Bitcoin dan sejumlah jaringan lainnya, karena tahan terhadap berbagai bentuk eksploitasi dan efisien dalam pemrosesan data. Hal ini juga diperkuat oleh rekomendasi dari banyak publikasi ilmiah dan standar industri, termasuk MDPI, yang menekankan keseimbangan optimal antara keamanan tinggi dan performa komputasi dalam sistem berbasis blockchain [15].

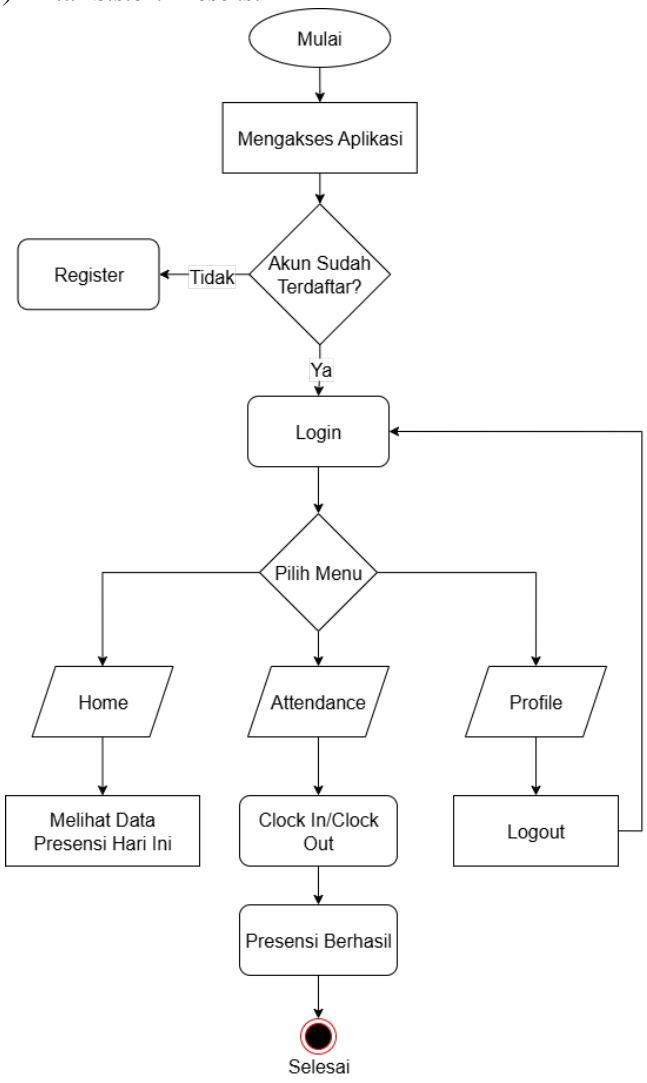
4) Proses Penambahan dan Replikasi Blok



Gbr. 4 Proses Penambahan dan Replikasi Blok

Ilustrasi pada Gbr. 4 menggambarkan proses penambahan dan replikasi blok menggunakan jaringan *peer-to-peer* (P2P), di mana pengguna mengajukan permintaan presensi yang kemudian diproses oleh server melalui proses *hashing*. Setelah proses hashing selesai, blok yang dihasilkan dikirimkan ke *node (peer)* lainnya untuk divalidasi. Setelah validasi berhasil, blok tersebut dikirim ke *node* lain untuk memastikan bahwa semua *node* memiliki salinan yang sama. Selanjutnya, distribusi blok dilakukan secara desentralisasi melalui jaringan blockchain melalui protokol komunikasi *real-time* WebSocket. *Node* yang pertama kali membuat atau menerima blok baru akan menyiarkan blok tersebut ke *node* lain untuk validasi. Jika data valid, blok disimpan dalam *database node* lokal dan kemudian disebarkan ke *node* lainnya. Jika data tidak valid, data tidak diproses dan tidak disimpan di *database*. Setelah proses ini selesai, salinan blok yang konsisten dibuat untuk seluruh jaringan, yang menjaga integritas dan keamanan data sistem. Untuk menjamin distribusi blok yang cepat dan lancar di seluruh jaringan, penting untuk menerapkan komunikasi P2P ini.

5) Alur Sistem Presensi



Gbr. 5 Alur Sistem Presensi

Ilustrasi Gbr. 5 ini menggambarkan alur system presensi yang dimulai dimulai ketika pengguna mengakses aplikasi, kemudian sistem akan mengecek apakah pengguna sudah memiliki akun. Jika belum, pengguna diarahkan untuk melakukan pendaftaran terlebih dahulu. Setelah terdaftar, pengguna dapat melakukan *login* untuk masuk ke dalam aplikasi. Setelah *login*, pengguna akan memilih menu yang tersedia, yaitu *home, attendance*, atau *profile*. Pada menu *home*, pengguna dapat melihat data presensi hari ini. Di menu attendance, pengguna bisa melakukan *clock in* atau *clock out*, dan jika berhasil, akan muncul notifikasi bahwa presensi berhasil. Dalam proses *clock in* dan *clock out*, integrasi blockchain terjadi. Sementara itu, pada menu *profile*, pengguna dapat melihat informasi profil serta melakukan *logout*.

C. Develop

Setelah tahap *Design* sistem selesai, maka proses selanjutnya ialah tahap *Develop*. Pada tahap ini, penulis

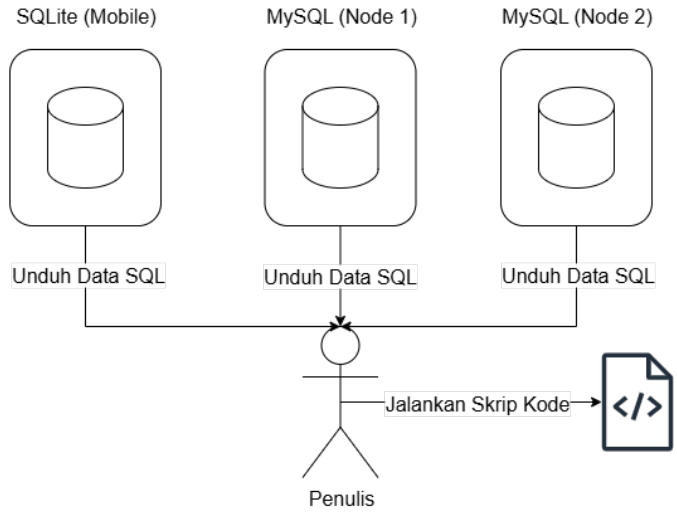
melakukan implementasi dari berbagai fitur dan fungsionalitas yang telah ditentukan sebelumnya dengan menggunakan teknologi dan bahasa pemrograman yang sesuai.

D. Testing

Dalam penelitian ini, skenario pengujian sistem akan dinilai berdasarkan dua indikator utama, diantaranya:

1) Uji Validitas

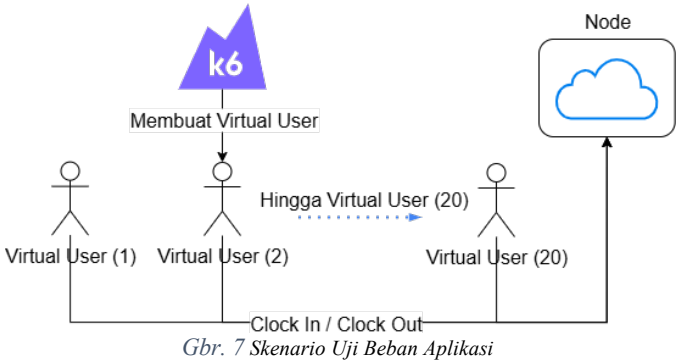
Pengujian ini dilakukan untuk memverifikasi validitas data presensi pada aplikasi mobile dengan menggunakan SQLite dan MySQL. Variabel pengujian yang dibutuhkan adalah data presensi dari pengguna berupa tanggal dan waktu saat *clock in* atau *clock out*. Berikut ini merupakan rincian dari skenario uji validitas:



Gbr. 6 Skenario Uji Validitas

Pada Gbr. 6, menunjukkan bahwa data presensi dari pengguna nantinya akan disimpan di 2 *node*, yakni SQLite (*database* mobile) dan MySQL (*database* server). Kemudian, penulis akan mengambil data presensi tersebut untuk dilakukan uji validitas datanya melalui *code script* yang telah penulis buat. Pengujian ini akan menghasilkan data valid dan tidak valid.

2) Uji Beban Aplikasi



Gbr. 7 Skenario Uji Beban Aplikasi

Pada Gbr. 7 ini merupakan skenario dari uji beban aplikasi. Tujuan dari pengujian ini adalah untuk mengevaluasi kemampuan aplikasi untuk menangani banyak permintaan sekaligus. Uji beban ini dilakukan menggunakan alat bantu Grafana K6. Variabel pengujian pada uji beban ini adalah 20 *virtual users*, RPS (*request per second*), HTTP *request duration* (*mean, max, median, min*), serta total *request* berhasil dan gagal. Berikut ini merupakan rincian dari skenario uji beban:

Proses pengujian beban ini dimulai dari k6 membuat *virtual user* sebanyak 20 *virtual users*. Lalu masing-masing *virtual user* akan mensimulasikan proses *clock in* atau *clock out* secara otomatis menggunakan Grafana k6 ke server atau *node* yang telah disiapkan khusus untuk uji coba.

Untuk mendapatkan hasil latensi rata-rata (*avg*) total, maka diperlukan formula sebagai berikut: [16].

$$\frac{\sum_{i=1}^{n} \times (avg_i \times jumlah\ request_i}{\sum_{1=1}^{n} \times jumlah\ request_i)}$$

TABEL I
KETERANGAN FORMULA

Keterangan	
$\sum_{i=1}^{n}$	Notasi sigma yang menyatakan penjumlahan dari iterasi ke-1 sampai ke-n
n	Jumlah total iterasi pengujian
avg_i	Nilai rata-rata (*average*) latensi dalam milidetik pada iterasi ke-i
$jumlah\ request_i$	Jumlah total permintaan (*requests*) yang berhasil pada iterasi ke-i

Berdasarkan penelitian dalam jurnal-jurnal terkait [17], batas wajar latensi sistem yang masih dapat diterima oleh pengguna adalah maksimal 250 milidetik (ms). Jika melebihi batas tersebut, keterlambatan mulai terasa dan dapat mengganggu interaksi pengguna, terutama pada sistem dengan arsitektur *thin client*. Dengan demikian, latensi kurang dari 250 ms tergolong baik dan optimal.

Setelah pengujian selesai, hasil dari setiap pengujian didokumentasikan dalam bentuk tabel dan grafik. Dokumentasi ini memberikan gambaran analisis yang lebih jelas dan informatif mengenai performa sistem.

E. Peluncuran Aplikasi

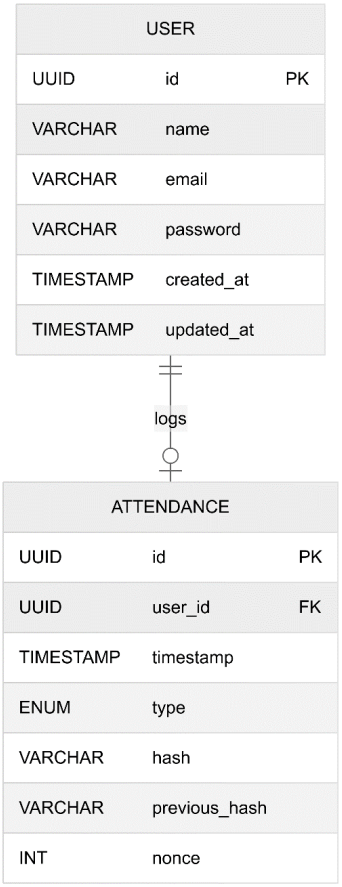
Pada tahap ini, aplikasi akan di-*deploy* ke VPS. Tujuannya adalah untuk mengisolasi node yang akan berjalan.

III. HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil dari proses pengembangan sistem presensi berbasis Android yang terintegrasi dengan teknologi blockchain. Dalam menggali kebutuhan sistem, penulis telah melakukan teknik elisitasi analisis dokumen. Dari analisis beberapa dokumen tersebut, maka dapat diidentifikasi beberapa kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional yang didapat ialah seperti fitur *register, login*, presensi, riwayat presensi, hingga integrasi blockchain. Selain itu, juga telah didapatkan kebutuhan non-fungsional yang mencakup aspek *security*, *scalability*, dan *reliability*. Berdasarkan hasil analisis tersebut, dirancanglah sistem dengan menggunakan *Entity Relationship Diagram (ERD)* untuk menggambarkan struktur data, serta *Data Flow Diagram (DFD)* untuk memvisualisasikan alur data pada sistem. Rancangan ini kemudian dijadikan dasar dalam

proses implementasi sistem, termasuk penerapan blockchain sebagai mekanisme keamanan data presensi. Selanjutnya, bab ini juga membahas tampilan antarmuka pengguna dan hasil pengujian sistem dari segi validitas data, performa beban, dan juga fungsionalitas sistem.

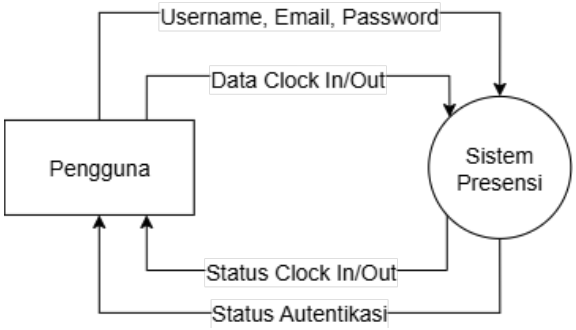
A. *Entity Relationship Diagram (ERD)*



Gbr. 8 ERD Aplikasi

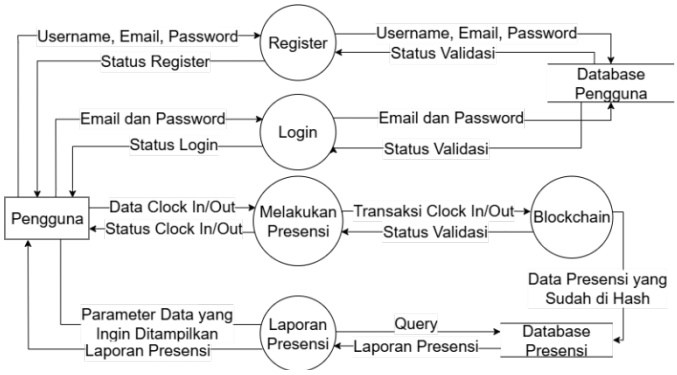
Diagram ERD pada Gbr. 8 ini menggambarkan sistem presensi yang mencatat kehadiran pengguna dengan mekanisme keamanan blockchain. Pada diagram ERD ini, memiliki 2 entitas yakni entitas *USER* dan entitas *ATTENDANCE.* Antar entitas tersebut memiliki hubungan *one-to-many (1:M)*, dimana satu pengguna dapat memiliki beberapa catatan kehadiran, tetapi setiap catatan kehadiran hanya dimiliki oleh satu pengguna.

B. *Data Flow Diagram (DFD)*



Gbr. 9 DFD Level 0

DFD (Data Flow Diagram) level 0 pada Gbr. 9 tersebut menggambarkan alur sistem presensi berbasis blockchain secara umum. Diagram ini menunjukkan tiga entitas utama, yakni Pengguna, Sistem Presensi, dan Blockchain. Pengguna mengirimkan kredensial autentikasi serta data *clock in/out* ke sistem presensi.

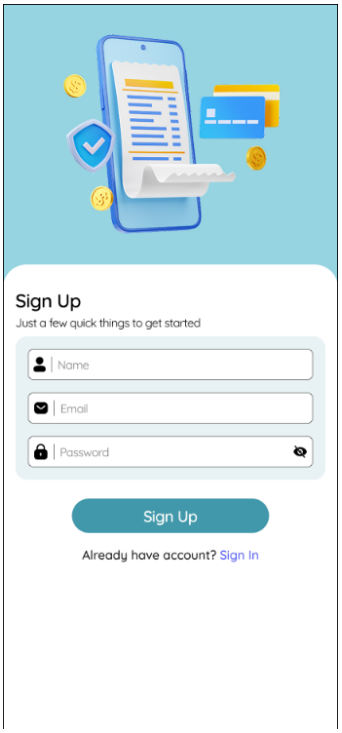


Gbr. 10 DFD Level 1

DFD (Data Flow Diagram) level 1 pada Gbr. 10 tersebut menggambarkan sistem presensi berbasis blockchain secara khusus. Diagram ini menjabarkan proses-proses internal yang terjadi dalam sistem presensi, dengan fokus pada validasi autentikasi, pengelolaan data *clock in/out*, dan integrasi dengan blockchain.

C. Tampilan Pengguna

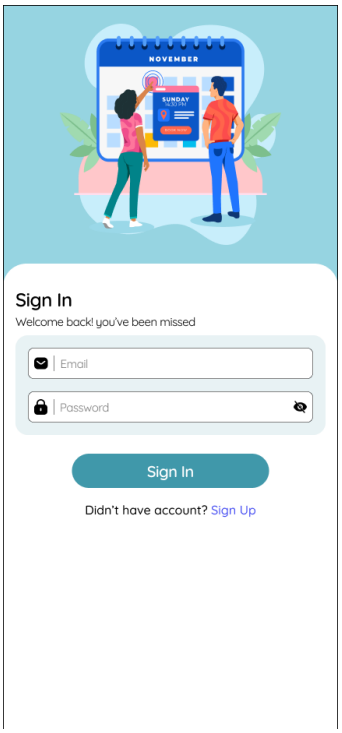
1) Sign Up



Gbr. 11 Sign Up

Pada Gbr. 11 ini menunjukkan halaman pendaftaran (*sign up*) yang digunakan oleh pengguna untuk membuat akun baru.

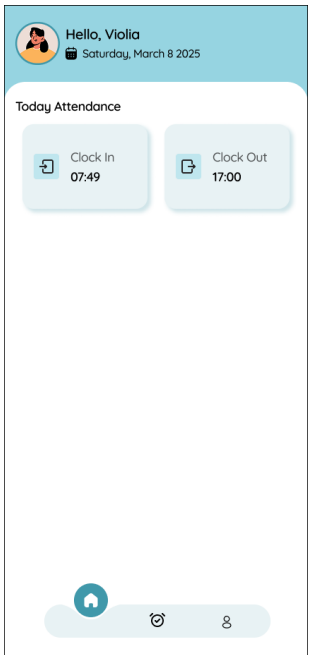
2) Sign In



Gbr. 12 Sign In

Pada Gbr. 12, menunjukkan halaman masuk (*sign in*) yang digunakan oleh pengguna untuk mengakses akun yang sudah terdaftar.

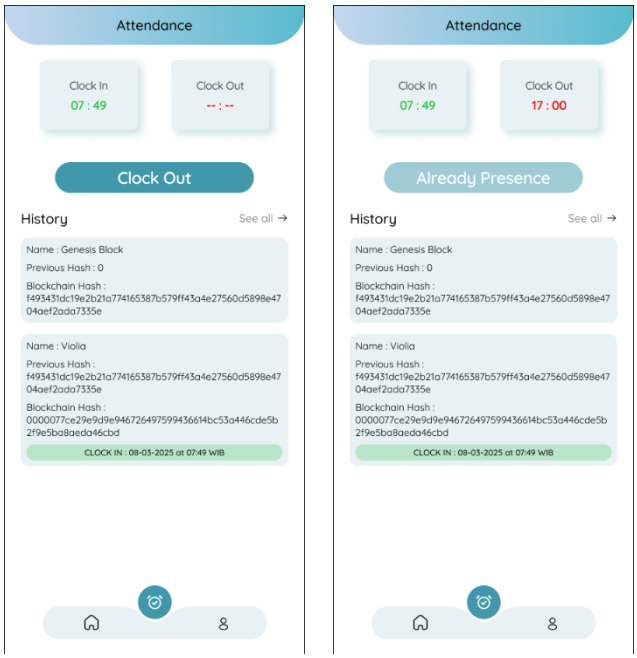
3) Home



Gbr. 13 Home

Pada Gbr. 13, menunjukkan halaman *home* yang menampilkan nama, tanggal, dan riwayat presensi hari ini.

4) Attendance



Gbr. 14 Berhasil Clock In Telah Melakukan Presensi Hari ini

Pada Gbr. 14, menunjukkan tampilan setelah pengguna berhasil melakukan presensi (*clock in*), sistem akan secara otomatis menampilkan waktu presensi dan memperbarui

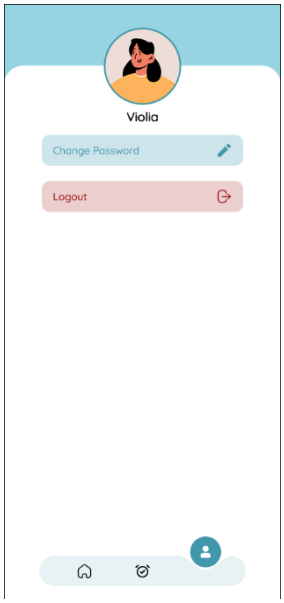
tampilan riwayat presensi dalam bentuk blockchain, yang mencakup informasi seperti nama, *previous hash*, *blockchain hash*, jenis presensi, serta tanggal dan waktu pelaksanaan. Tombol utama pada halaman *attendance* akan berubah menjadi tombol *clock out* sebagai respons terhadap aksi presensi yang telah dilakukan. Jika pengguna telah melakukan presensi lengkap (*clock in* dan *clock out)* pada hari yang sama, maka tombol presensi akan dinonaktifkan secara otomatis dan akan kembali aktif pada hari berikutnya.

5) History Detail

Gbr. 15 History Detail

Pada Gbr. 15, menunjukkan halaman *history detail* yang merupakan rincian riwayat presensi dari semua pengguna. Riwayat presensi yang tercatat yakni berupa nama, *previous hash*, blockchain *hash*, beserta tanggal dan waktu saat melakukan presensi.

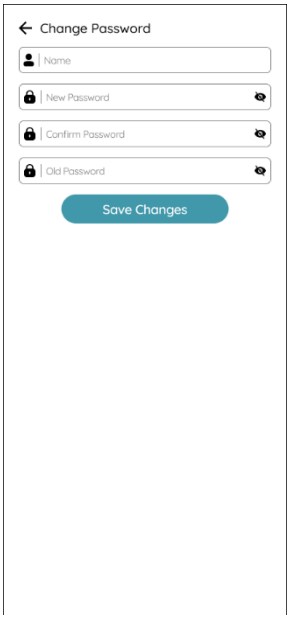
6) Profile



Gbr. 16 Profile

Pada Gbr. 16, menampilkan data pengguna berupa nama pengguna, menu untuk mengubah kata sandi, serta tombol *logout* yang berfungsi untuk keluar dari aplikasi.

7) Change Password



Gbr. 17 Change Password

Pada Gbr. 17, menampilkan menu untuk mengubah kata sandi pengguna. Selama proses ini, pengguna harus mengisi nama, *password* lama, dan *password* baru, serta konfirmasi *password* baru yang pengguna telah buat. Setelah data diisi dengan benar dan pengguna menekan tombol "*Save Changes*", sistem akan mengubah kata sandi sesuai dengan informasi yang dimasukkan.

8) Hasil Uji Coba Validitas Data

Pengujian validitas data dilakukan untuk memastikan bahwa data yang dicatat dan disimpan dalam dua sistem, yaitu *database* server (MySQL) dan *database* lokal pada perangkat mobile (SQLite). Pengujian ini penting untuk menjamin integritas data presensi yang direkam melalui aplikasi mobile dan kemudian disinkronkan ke server pusat. Berikut ini merupakan hasil running dari kode yang telah dijalankan:

TABEL II
DATA PRESENSI PADA DATABASE SERVER

id	userId	type	date	timestamp	hash	previousHash	nonce
1	1	GENESIS	2025-04-15	1744712854845	0	0	0
2	1454	CLOCK_IN	2025-04-15	1744712892386	003909950389e38021d78ee3628e656628a9f3416662d954bbbd6923b4f1c5bc	000000000000000000000000000000000000000000000000000000000000	130
3	9960	CLOCK_IN	2025-04-15	1744712892901	00e4436fae69468f322d99a9327349122a7e457d6bab123fa002b2ae73362f9c	003909950389e38021d78ee3628e656628a9f3416662d954bbbd6923b4f1c5bc	41
4	553	CLOCK_IN	2025-04-15	1744712892905	0032dc7594144e748c6e12e5b7e06fb16421710801908348fa5966e91a964d8d	00e4436fae69468f322d99a9327349122a7e457d6bab123fa002b2ae73362f9c	144
5	9303	CLOCK_IN	2025-04-15	1744712892913	00156dd6db631e342cfa8fbd6f54b01616dc306b8cec57af35d1510a5718a200	0032dc7594144e748c6e12e5b7e06fb16421710801908348fa5966e91a964d8d	37

TABEL III
DATA PRESENSI PADA DATABASE MOBILE

id	userId	type	date	timestamp	hash	previousHash	nonce
1	1	GENESIS	2025-04-15	1744712854845	0	0	0
2	1454	CLOCK_IN	2025-04-15	1744712892386	003909950389e38021d78ee3628e656628a9f3416662d954bbbd6923b4f1c5bc	000000000000000000000000000000000000000000000000000000000000	130
3	9960	CLOCK_IN	2025-04-15	1744712892901	00e4436fae69468f322d99a9327349122a7e457d6bab123fa002b2ae73362f9c	003909950389e38021d78ee3628e656628a9f3416662d954bbbd6923b4f1c5bc	41
4	553	CLOCK_IN	2025-04-15	1744712892905	0032dc7594144e748c6e12e5b7e06fb16421710801908348fa5966e91a964d8d	00e4436fae69468f322d99a9327349122a7e457d6bab123fa002b2ae73362f9c	144
5	9303	CLOCK_IN	2025-04-15	1744712892913	00156dd6db631e342cfa8fbd6f54b01616dc306b8cec57af35d1510a5718a200	0032dc7594144e748c6e12e5b7e06fb16421710801908348fa5966e91a964d8d	37

Tabel II dan III masing-masing menampilkan isi dari *database* server (MySQL) dan *database* lokal pada perangkat mobile (SQLite) yang berisi data aktivitas presensi pengguna, seperti *clock in* dan *clock out*. Validasi dilakukan dengan membandingkan beberapa variabel penting pada setiap entri data, yaitu *userId, type, date, timestamp, hash, previousHash*, dan *nonce*. Hasil perbandingan menunjukkan bahwa seluruh data pada kedua *database* tersebut identik, baik dari segi waktu maupun identitas pengguna. Hal ini mengindikasikan bahwa proses pencatatan, penyimpanan, dan sinkronisasi data berlangsung secara konsisten. Penggunaan nilai *hash* dan *previous hash* turut memperkuat integritas data, karena setiap perubahan akan berdampak pada struktur rantai *hash* yang terbentuk. Dengan demikian, sistem presensi yang dikembangkan terbukti mampu menyimpan dan mereplikasi data secara valid dan andal pada kedua *database* yang digunakan.

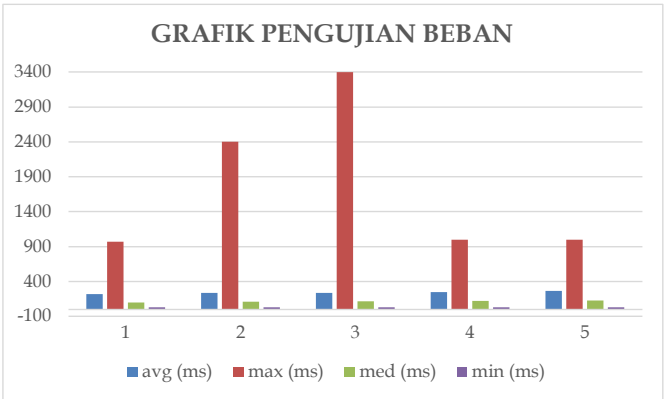
9) Hasil Uji Coba Beban Aplikasi

Pengujian beban aplikasi dilakukan menggunakan *tool* Grafana k6 serta skrip untuk melakukan presensi. Berikut ini merupakan hasil dari pengujian beban aplikasi yang ditunjukkan pada Tabel IV:

TABEL IV
HASIL PENGUJIAN ITERASI BEBAN APLIKASI

Iterasi	Avg (ms)	Max (ms)	Med (ms)	Min (ms)	Total	Success
1	217	967	98	27	9857	9857
2	237	2400	111	27	9701	9701
3	234	3400	115	27	9725	9725
4	247	1000	123	27	9621	9621
5	265	1000	128	27	9488	9488

Untuk mengukur performa dalam menangani permintaan secara bersamaan, pengujian beban sistem presensi dilakukan menggunakan alat Grafana k6 dengan 5 iterasi, masing-masing berlangsung 10 menit. Hasil pengujian menunjukkan bahwa seluruh permintaan berhasil diproses dengan tingkat keberhasilan 100%, yang menandakan sistem memiliki stabilitas yang baik. Rata-rata waktu respon berada dalam rentang 217 ms hingga 265 ms, tergolong responsif untuk aplikasi presensi. Meskipun terdapat lonjakan nilai maksimum pada iterasi ketiga sebesar 3400 ms, hal ini kemungkinan disebabkan oleh faktor eksternal seperti jaringan atau antrean transaksi. Waktu minimum tetap stabil di angka 27 ms pada seluruh iterasi, sedangkan waktu median menunjukkan tren kenaikan dari 98 ms menjadi 128 ms, yang masih berada dalam batas wajar untuk menunjang kenyamanan pengguna.



Gbr. 18 Grafik Pengujian Beban Aplikasi

Grafik batang pada Gbr. 18 menunjukkan perbandingan waktu respons berdasarkan nilai rata-rata, maksimum, median, dan minimum untuk setiap iterasi selama sepuluh menit.

Batang biru muda menunjukkan waktu rata-rata, batang merah menunjukkan waktu maksimum, batang hijau menunjukkan median, dan batang ungu menunjukkan waktu minimum. Visualisasi ini membuat lebih mudah untuk memahami variasi waktu respons untuk setiap pengukuran.

Berdasarkan penelitian dalam jurnal-jurnal terkait [17], batas wajar latensi sistem yang masih dapat diterima oleh pengguna adalah maksimal 250 milidetik (ms). Dengan demikian, latensi kurang dari 250 ms tergolong baik dan optimal. Berikut ini adalah rincian penghitungan total *avg* latensi sistem presensi berdasarkan rumus *Weighted Average* [16]:

$$Latensi\ Rata-Rata\ Total\ (ms) = \frac{\sum_{i=1}^{n} \times (avg_i \times jumlah\ request_i)}{\sum_{1=1}^{n} \times jumlah\ request_i)}$$

$$= \frac{(217 \times 9857) + (237 \times 9701) + (234 \times 9725) + (247 \times 9621) + (265 \times 9488)}{9857 + 9857 + 9725 + 9621 + 9488}$$

$$= \frac{11{,}604{,}463}{48{,}392} = 239{,}8\ ms$$

10) Hasil Uji Coba Fungsionalitas Aplikasi

Tujuan utama dari *fungsional testing* adalah untuk menguji apakah sistem tersebut melakukan apa yang seharusnya dilakukan, sesuai dengan kebutuhan dan harapan pengguna [18]. Dalam pengujian ini, penulis menguji 15 *test cases* yang masing-masing akan dijelaskan pada tabel-tabel berikut ini:

TABEL V
HASIL FUNCTIONAL TESTING FITUR AUTHORIZATION

No.	Test Case	Expected Output	Actual Output
1.	*Register* pengguna berhasil	Muncul toast "*Successfully register*" dan diarahkan langsung ke *home.*	Muncul toast "*Successfully register*" dan diarahkan langsung ke *home.*
2.	*Register* gagal karena email sudah digunakan	Muncul toast "*Email already exists*".	Muncul toast "*Email already exists*".
3.	*Register* gagal karena *password* terlalu pendek	Muncul toast "*Password must be at least 8 characters*".	Muncul toast "*Password must be at least 8 characters*".
4.	*Login* berhasil	Muncul toast "*Successfully login*" dan diarahkan langsung ke *home.*	Muncul toast "*Successfully login*" dan diarahkan langsung ke *home.*
5.	Login gagal karena pengguna tidak ditemukan	Muncul toast "*User not found*".	Muncul toast "*User not found*".
6.	Autentikasi berhasil dengan token valid	Validasi token sukses dan pengguna dikenali.	Validasi token sukses dan pengguna dikenali.
7.	Autentikasi gagal karena token invalid	Muncul toast "*Invalid token*" karena sistem berhasil menolak token tidak sah.	Muncul toast "*Invalid token*" karena sistem berhasil menolak token tidak sah.
8.	Autentikasi gagal karena token tidak diberikan	Muncul toast "*No token provided*" dan sistem memblokir permintaan tanpa token.	Muncul toast "*No token provided*" dan sistem memblokir permintaan tanpa token.

Hasil pengujian pada Tabel V menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur *authorization*, baik *register* maupun *login*. Pada modul *authentication* menunjukkan performa yang sesuai standar dengan validasi terhadap input pengguna, autentikasi berbasis token, serta penanganan kasus khusus seperti email yang telah terdaftar atau *password* yang tidak memenuhi syarat.

TABEL VI
HASIL FUNCTIONAL TESTING FITUR ATTENDANCE

No.	Test Case	Expected Output	Actual Output
1.	Mendapat-kan daftar presensi	Berhasil menampilkan daftar presensi.	Permintaan berhasil, respons sesuai dengan yang diharapkan.
2.	Mendapat-kan presensi terakhir pengguna	Berhasil menampilkan daftar presensi terakhir pengguna.	Berhasil menampil-kan daftar presensi terakhir pengguna.
3.	Melakukan *clock in*	Muncul toast "*Clock in successfully*".	Proses *clock in* berhasil dan blok baru ditambang (*mining*).
4.	Melakukan *clock out*	Muncul toast "*Clock out successfully*".	Proses *clock out* berhasil dan blok baru ditambang (*mining*).

Hasil pengujian pada Tabel VI menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur pencatatan kehadiran baik untuk *clock in* maupun *clock out* berhasil dilakukan, serta data kehadiran dapat diambil dengan benar melalui *endpoint* yang tersedia.

TABEL VII
HASIL FUNCTIONAL TESTING FITUR PROFILE

No.	Test Case	Expected Output	Actual Output
1.	Mendapat-kan profil pengguna	Menampilkan informasi profil.	Menampilkan informasi profil.
2.	Memper-barui profil pengguna	Berhasil memperbarui profil pengguna.	Berhasil memperbarui profil pengguna.
3.	Gagal mendapat-kan profil tanpa autenti-kasi	Muncul toast "*Unauthorized*" dan akses ditolak sesuai prosedur autentikasi token.	Muncul toast "*Unauthorized*" dan akses ditolak sesuai prosedur autentikasi token.

Hasil pengujian pada Tabel VII menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur

profile. Pada modul *profile* juga berhasil memperbarui dan menampilkan data pengguna sesuai permintaan, serta memberikan respons yang tepat ketika pengguna tidak memiliki otorisasi.

Berdasarkan hasil pengujian yang dilakukan terhadap sistem menggunakan 15 skenario uji yang mencakup modul *attendance, authentication,* dan *profile*, seluruh *test case* berhasil dijalankan dengan hasil sesuai ekspektasi. Hal ini menunjukkan bahwa seluruh fitur inti dari aplikasi telah berjalan dengan baik, tanpa adanya kegagalan fungsi atau *error* sistem selama proses pengujian berlangsung.

IV. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem presensi berbasis Android yang terintegrasi dengan teknologi blockchain guna meningkatkan keamanan, keandalan, dan transparansi data kehadiran. Sistem yang dikembangkan menggunakan *framework* Flutter pada sisi antarmuka serta Elysia.js, MySQL, Redis, dan algoritma SHA-256 pada sisi *backend*, membentuk struktur blok presensi yang bersifat aman dan tidak dapat diubah (*immutable*).

Hasil pengujian menunjukkan bahwa data presensi antara *database* lokal (SQLite) dan *database* server (MySQL) bersifat konsisten dan sinkron, dengan atribut *seperti userId, type, timestamp, hash*, dan *nonce* tercatat secara identik. Uji performa beban aplikasi dengan menggunakan 20 *virtual users* menunjukkan performa yang baik, dengan rata-rata latensi 239,8 ms karena masih dalam batas wajar (<250 ms) dengan tingkat keberhasilan pemrosesan permintaan mencapai 100%. Selain itu, pengujian fungsionalitas terhadap 15 skenario membuktikan bahwa seluruh fitur berjalan sesuai dengan kebutuhan. Dengan demikian, integrasi teknologi blockchain dalam sistem presensi mobile terbukti efektif dalam menjaga integritas, validitas, dan keandalan data, serta layak untuk diadopsi dalam pengembangan sistem presensi digital yang aman dan transparan.

REFERENSI

[1] I. Wijaya, E. Haryatmi, dan A. B. Kurniawan, "InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan Attribution-NonCommercial 4.0 International. Some rights reserved Blockchain Implementasi Teknologi Blockchain pada Sistem Presensi Staff VM LePKom Berbasis Web," vol. 5, no. 1, 2020, doi: 10.30743/infotekjar.v5i1.2932.

[2] N. Erzed, N. Anwar, A. M. Widodo, E. Prasetyo, dan K. K. Juman, "Implementasi Flutter Pada Aplikasi Presensi Karyawan Berbasis Mobile." [Daring]. Tersedia pada: https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive

[3] T. R. Gadekallu *dkk.*, "Blockchain for Edge of Things: Applications, Opportunities, and Challenges," Okt 2021, [Daring]. Tersedia pada: http://arxiv.org/abs/2110.05022

[4] R. Kurniawati, A. A. Rizky, dan A. Hermawan, "Implementasi Smart Device untuk Sistem Presensi Perkuliahan," *Jurnal Manajemen Informatika (JAMIKA)*, doi: 10.34010/jamika.v10i1.

[5] I. Meirobie, A. P. Irawan, H. T. Sukmana, D. P. Lazirkha, dan N. P. L. Santoso, "Framework Authentication e-document using Blockchain Technology on the Government system," *International Journal of Artificial Intelligence Research*, vol. 6, no. 2, Jul 2022, doi: 10.29099/ijair.v6i2.294.

[6] P. R. Setiawan, R. A. Ramadhan, D. A. Labellapansa, P. Koresponden, : Panji, dan R. Setiawan, "Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan Pelatihan Pemrograman Flutter."

[7] R. F. Ramadhan dan R. Mukhaiyar, "Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi," 2020.

[8] [8] SQLite, "SQLite — sqlite.org," https://www.sqlite.org/ [Accessed 20-12-2024].

[9] M. A. Kausar, M. Nasar, M. Abu Kausar, dan A. Soosaimanickam, "INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY A Study of Performance and Comparison of NoSQL Databases: MongoDB, Cassandra, and Redis Using YCSB," *Abu Kausar et al. / Indian Journal of Science and Technology*, vol. 15, no. 31, hlm. 1532–1540, 2022, doi: 10.17485/IJST/v15i31.1352.

[10] Grafana. Labs, "Grafana — grafana.com," https://grafana.com/docs/k6/latest/, [Accessed 19-12- 2024].

[11] Y. E. Oktian, "Design and Implementation of Blockchain-Based Office Attendance System," *Teknika*, vol. 13, no. 1, hlm. 137–144, Mar 2024, doi: 10.34148/teknika.v13i1.775.

[12] M. Jobair, H. Faruk, S. Subramanian, H. Shahriar, M. Valero, dan X. Li, "Software Engineering Process and Methodology in Blockchain-Oriented Software Development: A Systematic Study," 2022. [Daring]. Tersedia pada: https://www.researchgate.net/publication/359081547

[13] R. Salama dan M. H. Zahid, "Making sure of students attendance and the system of mark management by using mobile applications," *Global Journal of Computer Sciences: Theory and Research*, vol. 11, no. 1, hlm. 24–44, Apr 2021, doi: 10.18844/gjcs.v11i1.5383.

[14] S. Ar dan B. Gupta Banik, "A Comprehensive Study of Blockchain Services: Future of Cryptography," 2020. [Daring]. Tersedia pada: www.ijacsa.thesai.org

[15] A. Sevin dan A. A. Osman Mohammed, "Comparative Study of Blockchain Hashing Algorithms with a Proposal for HashLEA," *Applied Sciences (Switzerland)*, vol. 14, no. 24, Des 2024, doi: 10.3390/app142411967.

[16] D. S. Voss, "Aggregation," *Encyclopedia of Social Measurement, Three-Volume Set*, vol. 1, hlm. 33–42, Jan 2005, doi: 10.1016/B0-12-369398-5/00044-X.

[17] C. Attig, N. Rauh, T. Franke, dan J. F. Krems, "System latency guidelines then and now – Is zero latency really considered necessary?," dalam *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2017, hlm. 3–14. doi: 10.1007/978-3-319-58475-1_1.

[18] "FunctionalSoftware".