

Implementasi Teknologi *Blockchain* dalam Pengembangan Aplikasi *Web* Terdesentralisasi untuk Pengelolaan Data Pos Pelayanan Terpadu (Studi Kasus: Posyandu Mawar Lingkungan Gibug)

Nurhadi Qomaruddin¹, Yana Aditia Gerhana², Ichsan Taufik³,
Cepy Slamet⁴, Muhammad Deden Firdaus⁵

^{1,2,3,4,5}Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia

Article Info

Article history:

Received
Revised
Accepted
Published

Keywords:

Blockchain
Data Management
Decentralized Application
Posyandu
Smart Contract

ABSTRACT

The Integrated Service Post (Posyandu) is a community-based health service established by the government, playing an important role in monitoring child health, including efforts to reduce infant and child mortality rates. However, data management at Posyandu is generally still conducted manually using paper-based records, making it prone to data loss and inefficient in terms of access and tracking. One common approach to overcoming these challenges is the use of distributed data systems, which allow data storage and processing to occur across multiple computers in different locations. Nevertheless, many of these systems still rely on centralized servers, making them vulnerable to data breaches and manipulation due to the single point of storage. To address this issue, this research proposes the development of a decentralized web application based on blockchain technology as a solution for secure, transparent, and traceable data management. The application is developed using smart contracts written in Solidity, deployed on the Ethereum blockchain, with Hardhat as the backend framework and React.js as the user interface. The system was developed using a prototyping methodology and evaluated through black-box testing to assess its functional performance. Test results show that the application is capable of managing data effectively, while maintaining a high level of security and transparency. By adopting blockchain technology, the system enhances the effectiveness and efficiency of Posyandu's data management, while ensuring data integrity and traceability within a decentralized environment.

Corresponding Author:

Nurhadi Qomaruddin,
Informatics Department, Faculty of Science & Technology, UIN Sunan Gunung Djati Bandung
Jl. A. H. Nasution No. 105, Cibiru, Bandung, Indonesia. 40614
Email: join@uinsgd.ac.id

1. PENDAHULUAN

Teknologi saat ini telah mengalami banyak perkembangan, terlebih dalam segi keamanan, kekekalan data, dan transparansi. Masalah yang terjadi pada masa kini adalah banyaknya sistem informasi yang tidak memanfaatkan perkembangan teknologi yang telah disebutkan, yaitu dengan membuatnya terpusat pada satu penyimpanan data yang menyebabkan rentannya keamanan data pada sistem informasi tersebut. Hal ini menjadi kritis terutama pada sistem informasi publik yang bersifat jangka panjang dan menyangkut data penting, seperti data pertumbuhan anak di pos pelayanan terpadu atau lebih dikenal dengan Posyandu.

Posyandu yang merupakan sebuah layanan kesehatan yang dibentuk oleh pemerintah sebagai media untuk menurunkan angka kematian bayi dan anak balita, serta memiliki peran dalam

menanggulangi masalah kesehatan masyarakat, seperti masalah *stunting*. Namun pada umumnya, dalam proses pencatatan data di posyandu masih menggunakan pencatatan secara manual dengan menulis di buku catatan. Hal ini menyulitkan kader posyandu dalam mengelola dan menyimpan data. Di samping itu rawannya kehilangan data karena lembaran dari buku pencatatan posyandu yang tercecer.

Di samping itu, ibu balita membutuhkan sebuah media untuk memberikan informasi status kesehatan dan perkembangan anaknya kapan saja dan di mana saja. Salah satu pendekatan yang umum digunakan adalah sistem data terdistribusi, yang memungkinkan penyimpanan dan pemrosesan data dilakukan oleh beberapa komputer pada lokasi yang berbeda. tetapi pada umumnya sistem data terdistribusi dikelola secara terpusat dengan menyimpan data secara sentral, hal ini menyebabkan rentannya terhadap serangan tunggal, sehingga orang mudah mencuri atau memanipulasi data secara tidak bertanggungjawab[1].

Maka dari itu, teknologi *Blockchain* menawarkan solusi berupa sistem informasi yang tidak terpusat atau terdesentralisasi yang mampu menjamin keamanan, keterlacakan, transparansi, dan kekekalan data. Setiap data yang ditulis pada *Blockchain* bersifat *immutable* (tidak dapat diubah atau dihapus), sehingga sangat cocok untuk sistem yang membutuhkan keandalan tinggi. *Smart contract* sebagai bagian dari *Blockchain* yang membuat pengelolaan dan penyimpanan data secara terdesentralisasi sehingga tidak berpusat pada satu *server*.

Sehingga bagi posyandu, teknologi *Blockchain* membuat setiap catatan kunjungan dan kesehatan anak akan terenskripsi dan disimpan dalam *Blockchain*, hal ini membuat sistem memiliki keamanan data yang terjamin, karena akan sulit sekali data untuk diubah ataupun diretas. Di samping itu, karena karakteristik dari *Blockchain* itu sendiri bersifat transparan yang membuat setiap transaksi yang dilakukan dapat diketahui oleh siapa transaksi itu dilakukan dan transaksi apa yang dilakukannya.

Penelitian ini menggunakan studi kasus posyandu Mawar yang berlokasi di Lingkungan Gibug, Desa Cigadung, Kecamatan Cigugur, Kabupaten Kuningan, Jawa Barat. Alasan peneliti memilih posyandu tersebut sebagai studi kasus, karena pertimbangan peneliti perihal posyandu ini yang masih melakukan pencatatan secara manual di buku catatan, hal ini sejalan dengan masalah penelitian yang diangkat oleh peneliti. Di samping itu, alasan lain mengapa peneliti menggunakan studi kasus tersebut adalah mudahnya akses untuk keperluan data penelitian.

Berdasarkan penelitian Sri Kusumadewi, dkk(2019) menyatakan bahwa dengan penggunaan aplikasi teknologi informasi dapat meningkatkan kualitas dari kader posyandu dalam memberikan pelayanan kepada masyarakat. Pada pengaplikasiannya, peneliti menggunakan platform *web* dan *android* yang akhirnya diketahui bahwa aplikasi sistem informasi posyandu berbasis *android* lebih mudah dioperasikan dibandingkan dengan aplikasi berbasis *web* karena mayoritas kader posyandu lebih terampil menggunakan *smartphone* dibandingkan[2].

Adapun penelitian dari Nadia (2023) menyatakan bahwa teknologi *Blockchain* telah meningkatkan keamanan data dengan mengurangi risiko manipulasi data dan peretasan dalam sistem terdistribusi pada organisasi X, dimana keberhasilannya dipengaruhi oleh beberapa faktor seperti strategi keamanan yang diterapkan, kesesuaian teknologi dengan yang dibutuhkan organisasi, dan arsitektur *Blockchain*[3]. Hal ini sejalan penelitian Setiawan Restu Aji dengan pernyataan yang menyatakan bahwa teknologi *Blockchain* menunjukkan potensi besar dalam meningkatkan keamanan dan integritas pemilihan[4].

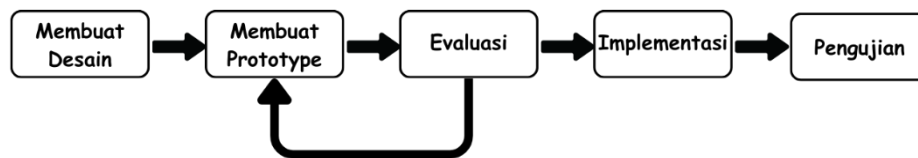
Berbeda dengan penelitian-penelitian sebelumnya yang membangun aplikasi posyandu dengan menyimpan data terpusat pada 1 *server*, penelitian ini membangun aplikasi posyandu secara terdesentralisasi. Adapun penelitian lainnya tentang pembangunan aplikasi *blockchain* yang mana pada penelitian-penelitian tersebut hanya bisa memberikan masukkan ke dalam aplikasi *Blockchain* tanpa bisa melakukan eksekusi lainnya terhadap data yang ada, akan tetapi dalam penelitian ini, aplikasi dirancang untuk dapat melakukan berbagai eksekusi seperti *create*, *read*, *update*, *delete* terhadap data yang tersimpan dalam *Blockchain*. Hal ini membuat *user* yang memiliki akun *wallet* dan terdaftar sebagai kader posyandu dapat melakukan interaksi atau eksekusi pada aplikasi *Blockchain* dengan menggunakan gas *fee* sebagai pembayarannya.

2. METODOLOGI

Metode yang digunakan pada pengumpulan data dalam pengembangan aplikasi ini adalah wawancara dan observasi. Menurut pengertiannya, wawancara adalah komunikasi antara dua pihak atau lebih yang bisa dilakukan dengan tatap muka di mana salah satu pihak berperan sebagai

interviewer dan pihak lainnya berperan sebagai *interviewee* dengan tujuan tertentu, misalnya untuk mendapatkan informasi atau mengumpulkan data. Sedangkan observasi merupakan proses pengamatan sistematis dari aktivitas manusia dan pengaturan fisik dimana kegiatan tersebut berlangsung secara terus menerus dari lokus aktivitas bersifat alami untuk menghasilkan fakta[5]. Subjek dari wawancara ini adalah kader posyandu dari Posyandu Mawar lingkungan Gibug, Desa Cigadung. Sedangkan observasi dilakukan secara langsung ke Desa Cigadung untuk mendapatkan kumpulan data yang diperlukan dalam mengembangkan aplikasi ini. Berdasarkan permasalahan yang telah didefinisikan dari wawancara dan observasi, maka aplikasi berbasis *Blockchain* dapat menjadi solusi karena mampu menjamin keamanan, keterlacakan, transparansi, dan kekekalan data.

Pada pengembangan perangkat lunak penelitian ini, menggunakan metode *prototype* yang mana menurut pengertiannya *prototype* adalah versi awal dari sistem perangkat lunak yang digunakan untuk mendemonstrasikan konsep-konsep, percobaan rancangan, dan menemukan lebih banyak masalah dan solusi yang memungkinkan[6]. Metode ini memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna.



Gambar 1. Metode *Prototype*

Pada Gambar 1. Menggambarkan tahapan dari metode *prototype* yang digunakan pada penelitian ini, beberapa tahapan yang dilakukan pada penelitian ini yaitu, membuat desain, membuat *prototype*, evaluasi, implementasi dan pengujian

Pada pengujian perangkat lunak ini menggunakan metode pengujian black box yang mana pengujian black box merupakan adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Pengujian ini akan dijalankan oleh *software tester* yang mana pada penelitian ini adalah kader posyandu Mawar dari Desa Cigadung.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Penggunaan Teknologi Blockchain

3.1.1. Keamanan

Aplikasi ini menggunakan *smart contract* pada setiap transaksinya baik pada proses pencatatan maupun pembaruan data. Karena data disimpan dalam *blockchain* maka, tidak ada pihak yang dapat memanipulasi data secara tidak bertanggungjawab. Di samping itu, karena setiap proses transaksi hanya dapat dijalankan oleh *user* yang memiliki *wallet account* yang valid, maka keamanan data di aplikasi dapat terjamin.

```
require(users[msg.sender].registered, "User not registered");
```

Gambar 2. Code User Registered

Pada Gambar 2 memperlihatkan kode yang mana memastikan bahwa hanya *user* yang telah terdaftar yang dapat melakukan setiap proses transaksi.

3.1.2. Immutability (Kekekalan Data)

Karena blockchain memiliki karakteristik immutable atau tidak dapat diubah dan aplikasi ini mengimplementasikan *blockchain* pada pengembangannya, maka data pada aplikasi ini tidak dapat dihapus atau diubah permanen. Maka dari itu, setiap proses transaksi pada aplikasi ini, baik itu menghapus atau mengubah data, sebenarnya hanya menandai bahwa dihapus atau diubah pada *event*, dan tampilan antarmuka akan membaca *event* tersebut.

```
function deletePerson(uint _id, uint _nik) public {
    require(_id < nextId, "Item does not exist");
    rangrang[_id].nik = _nik;
    rangrang[_id].isDeleted = true;
    emit TransactionHistory(msg.sender, users[msg.sender].username, "delete", "person", _nik, block.timestamp);
}
```

Gambar 3. Code Delete

Pada Gambar 3 menjelaskan bahwa data pada aplikasi tidak sebenarnya diubah, hanya diberi tanda pada *event* yang akan dibaca oleh antarmuka pengguna.

3.1.3. Transparansi

Sifat lainnya dari *blockchain* adalah transparan, sehingga semua data dan transaksi dapat terlihat oleh siapa saja. Pada aplikasi ini, setiap proses transaksi yang dilakukan akan tercatat dan akan ditampilkan, sehingga setiap orang dapat mengetahui siapa yang telah melakukan transaksi dan transaksi apa yang dilakukan.

3.1.4. Terdesentralisasi

Aplikasi mengimplementasikan *blockchain* untuk pengembangannya sehingga aplikasi ini bersifat terdesentralisasi, dengan kata lain aplikasi ini tidak menyimpan data terpusat pada satu server. Hal ini membuat aplikasi ini terjamin keamanannya karena minim resiko akan pencurian dan manipulasi data secara tidak bertanggungjawab.

3.2. Implementasi Sistem

3.2.1. Implementasi Perangkat Lunak

Tabel 1. Implementasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 10
Bahasa Pemrograman	Javascript Ver. 22.13.1, Solidity Ver. 0.8.28
Code Editor	Visual Studio Code Ver. 1.99
Framework	Reactjs Ver. 19.0.0 & Hardhat Ver. 2.23.0
Dompet Digital (crypto wallet)	Metamask Ver.12.16.0
Browser	Google Chrome Ver. 135.0.7049.96

Pada Tabel 1 diperlihatkan mengenai implementasi perangkat lunak pada aplikasi. Pada implementasi perangkat lunak dalam pembuatan aplikasi ini menggunakan sistem operasi Windows 10 dengan bahasa pemrograman Solidity dan *framework* Hardhat untuk membuat *smart contract* serta menggunakan Metamask sebagai dompet digital. Dalam membuat *interface* menggunakan bahasa pemrograman Javascript dengan *framework* ReactJS. Code editor yang digunakan adalah Visual Studio Code dengan browser Google Chrome.

3.2.2. Implementasi Perangkat Keras

Tabel 2. Implementasi Perangkat Keras

Perangkat Keras	Spesifikasi
System Model	Vivobook X441MA
Processor	Intel Celeron N4000
Memory	4096 MB RAM
Storage	120 GB SSD & 1 TB HDD
VGA	Intel UHD Graphics 600
System Model	Vivobook X441MA

Pada Tabel 2 diperlihatkan mengenai implementasi perangkat keras pada aplikasi. Pada implementasi perangkat keras menggunakan *device* Asus Vivobook X441MA yang dilengkapi dengan processor Intel Celeron N4000 dengan penyimpanan 4GB RAM ,120 GB SSD dan 1TB HDD. VGA yang digunakan adalah Intel UHD Graphics 600.

3.2.3. Implementasi Blockchain

Pada Tabel 2 diperlihatkan mengenai implementasi perangkat keras pada aplikasi. Pada implementasi perangkat keras menggunakan *device* Asus Vivobook X441MA yang dilengkapi dengan

processor Intel Celeron N4000 dengan penyimpanan 4GB RAM ,120 GB SSD dan 1TB HDD. VGA yang digunakan adalah Intel UHD Graphics 600.

3.2.4. Implementasi Interface

Aplikasi ini menggunakan ReactJs sebagai *frontend* yang akan menampilkan *user interface*, serta aplikasi ini menggunakan ether.js untuk menghubungkan ReactJs dengan *smart contract* yang telah dibuat. Berbagai *interface* dibuat, di antaranya adalah ;

1. *Interface Landing Page*

Interface dari *landing page* menampilkan pilihan bagi *user* untuk masuk sebagai kader posyandu atau masuk sebagai ibu balita. Jika *user* memilih masuk sebagai kader posyandu, maka sistem akan mengarahkan *user* ke halaman *login* bagi kader posyandu. Sedangkan, jika *user* memilih untuk masuk sebagai ibu balita, maka sistem akan mengarahkan *user* ke halaman ibu balita.

2. *Interface Sign Up*

Interface sign up diperuntukkan kepada kader posyandu yang mana kader posyandu akan diminta untuk mendaftarkan *username* dan *password*. Pada halaman ini, dibutuhkan akun *wallet* untuk bisa mendaftar, jika *user* tidak memiliki akun *wallet* atau tidak menghubungkan akun *wallet*, maka registrasi akun akan gagal.

3. *Interface Login*

Interface login terbagi menjadi 2, yaitu *login* sebagai kader posyandu dan *login* sebagai ibu balita. Jika *user* berhasil *login* sebagai kader posyandu dengan memasukkan *password* yang sesuai, maka sistem akan mengarahkan *user* ke halaman home untuk kader posyandu yang mana di dalamnya dapat mengakses berbagai macam transaksi dalam mengolah data posyandu. Sedangkan jika *user* berhasil masuk sebagai ibu balita dengan memasukkan QR *code* dari balita, maka *user* akan diarahkan ke halaman dari detail balita untuk ibu balita.

4. *Interface Homepage*

Pada halaman *homepage*, diperlihatkan jumlah dari balita, banyaknya kunjungan, serta jumlah dari kader yang ada di posyandu Mawar. *User* dapat berpindah halaman dengan memilih opsi menu yang ada di *navigation bar*.

5. *Interface List Balita*

Pada halaman *list balita*, *user* akan diperlihatkan daftar dari balita yang terdaftar dengan sebagian informasi darinya. Pada setiap baris dari data balita, *user* diberikan 2 *button* yang digunakan untuk menghapus data dari balita dan untuk melihat detail informasi dari balita yang dipilih. Jika *user* memilih untuk menghapus balita, maka sistem akan menghilangkan data balita dari halaman tersebut, sedangkan jika *user* memilih untuk melihat detail balita, maka *user* akan diarahkan pada halaman detail balita untuk kader posyandu.

6. *Interface List Kunjungan*

Pada halaman *list kunjungan*, *user* akan diperlihatkan semua kunjungan posyandu yang pernah terjadi. Semua kunjungan tersebut akan diperlihatkan pada sebuah tabel yang memuat nama balita pada kunjungan posyandu tersebut serta hasil dari kegiatan posyandu.

7. *Interface Input dan Edit Balita*

Pada halaman *input balita*, *user* dapat memberikan input untuk mengisi NIK balita, nama balita, tanggal lahir balita, NIK orang tua, nama orang tua, nomor HP orang tua, alamat (RT). Sedangkan pada halaman *edit balita*, *user* dapat mengubah semua data dari balita dan data tersebut akan menggantikan data sebelumnya

8. *Interface Input dan Edit Kunjungan*

Pada halaman *input kunjungan* dapat memberikan input untuk mengisi nama balita yang diambil dari data balita, tanggal kunjungan, berat badan, tinggi badan, lingkar kepala, lingkar lengan, dan vitamin A. Dan pada halaman *edit kunjungan* *user* dapat mengubah data kunjungan yang dipilih dan data tersebut akan menggantikan data sebelumnya.

9. *Interface Detail Balita*

Pada halaman detail menampilkan seluruh informasi dari setiap balita baik itu biodata dari balita maupun setiap kunjungan yang telah dilakukan balita tersebut. Ada hal yang membedakan halaman detail bagi kader dengan ibu balita, yaitu halaman detail bagi kader memuat *button* untuk menghapus dan mengedit setiap kunjungan dari balita, di mana jika *user* memilih opsi untuk hapus kunjungan, maka data kunjungan yang dari balita akan dihilangkan, serta jika *user* memilih untuk mengedit kunjungan yang terpilih, maka *user* akan diarahkan pada halaman edit

kunjungan. Sedangkan halaman balita bagi ibu posyandu hanya bisa dilihat tanpa bisa melakukan interaksi apapun di dalamnya.

10. *Interface Edukasi*

Pada halaman edukasi, ibu balita dapat melihat berbagai edukasi mengenai pencegahan *stunting*.

3.3 *Pengujian Sistem*

Tabel 3. Pengujian Black Box

<i>Test Class</i>	<i>Test Item</i>	<i>Respon diharapkan</i>	<i>Description</i>
Halaman <i>Login Kader</i>	Memasukkan <i>password</i> dari akun yang sudah terdaftar	Aplikasi dapat menerima <i>password</i> dan mengarahkan <i>user</i> ke <i>home</i> jika <i>password</i> benar.	Sukses, aplikasi berhasil menerima <i>password</i> dan mengarahkan <i>user</i> ke halaman <i>home</i> saat <i>user</i> memasukkan <i>password</i> yang benar.
Halaman <i>Login Ibu Balita</i>	Memasukkan QR code milik balita di posyandu Mawar.	Aplikasi dapat menerima input QR code dan mengarahkan ke halaman biodata anak jika QR code tersebut sesuai.	Sukses. Aplikasi berhasil membaca QR code yang dimasukkan dan mengarahkan ke halaman dari biodata anak yang telah dimasukkan QR codenya.
Halaman <i>Sign Up</i>	Memasukkan <i>username</i> dan <i>password</i> untuk mendaftar akun.	Aplikasi dapat mendaftarkan akun dari sebuah akun <i>wallet</i> dengan menerima input <i>username</i> serta <i>password</i> dan disimpan di <i>blockchain</i> .	Sukses. Aplikasi berhasil mendaftarkan akun dari sebuah akun <i>wallet</i> dan disimpan di <i>blockchain</i> .
Halaman <i>List Balita</i>	Melihat daftar dari balita, menghapus data balita, serta menekan tombol detail	Aplikasi dapat menampilkan data balita, dapat menghapus data balita, serta dapat menggunakan tombol detail yang akan mengarahkan <i>user</i> ke halaman detail balita.	Sukses. Aplikasi dapat menampilkan data balita yang ada di <i>Blockchain</i> , dapat menghapusnya serta dapat mengakses halaman detail balita.
Halaman <i>List Kunjungan</i>	Melihat semua daftar kunjungan dari balita.	Aplikasi dapat menampilkan semua kunjungan posyandu.	Sukses. Aplikasi dapat menampilkan setiap kunjungan yang sudah dimasukkan.
Halaman Input Balita	Memasukkan data balita.	Aplikasi dapat menerima input data balita dan menyimpannya di <i>blockchain</i> .	Sukses. Aplikasi berhasil menyimpan data balita ke dalam <i>blockchain</i> .
Halaman Input Kunjungan	Memasukkan data kunjungan.	Aplikasi dapat menerima input data	Sukses. Aplikasi dapat menyimpan

<i>Test Class</i>	<i>Test Item</i>	<i>Respon diharapkan</i>	<i>Description</i>
		kunjungan dan menghubungkannya dengan data balita yang melakukan kunjungan tersebut dengan memilih nama balita.	data kunjungan dan terhubung dengan data balita yang dipilih.
Halaman Dashboard	Melihat jumlah balita, kunjungan dan kader posyandu.	Aplikasi dapat menampilkan jumlah balita yang terdaftar, jumlah catatan kunjungan dari setiap balita dan jumlah kader Posyandu Mawar.	Sukses. Aplikasi dapat menampilkan jumlah balita, kunjungan dan kader.
Halaman Edukasi	Melihat informasi dan edukasi terkait <i>stunting</i> .	Aplikasi menampilkan halaman yang memuat informasi dan edukasi terkait <i>stunting</i> .	Sukses. Aplikasi dapat menampilkan informasi dan edukasi terkait <i>stunting</i> .

Pada Tabel 3 menjelaskan bahwa aplikasi setelah diuji dengan menggunakan metode black box memberikan hasil yang baik, yang mana aplikasi dapat memberikan respon yang sesuai dengan target yang telah ditentukan sebagai indikator keberhasilan pada aplikasi, pengujian ini dilakukan terhadap 9 *test item* yang berbeda.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa pengembangan aplikasi web terdesentralisasi untuk pengelolaan data Posyandu dengan teknologi *blockchain* dapat dilakukan dengan menghubungkan *smart contract* dengan *user interface*. *Smart contract* yang berperan sebagai backend dibuat dengan Bahasa Solidity dan *framework* Hardhat, kemudian dihubungkan ke Metamask sebagai dompet digital serta React.js sebagai *user interface*.

Hasil pengujian aplikasi menggunakan metode black box menunjukkan bahwa fungsionalitas aplikasi dapat berjalan dengan baik, Di mana aplikasi dapat merespon dengan baik setiap *test item* yang diberikan dan berjalan sesuai dengan target yang telah ditentukan sebagai indikator keberhasilan pada aplikasi. Di samping itu, aplikasi dengan mengimplementasikan *blockchain* di dalamnya, terbukti dapat meningkatkan keamanan, memiliki kekekalan data, dan bersifat transparan.

Keterbatasan pada penelitian ini adalah tingginya biaya *gas fee* dalam setiap proses transaksi yang dilakukan. Penelitian selanjutnya disarankan untuk menggunakan jaringan Polygon dalam menjalankan *smart contract*, selain karena jaringan polygon mengonsumsi sedikit *gas fee*, Polygon juga telah terintegrasi dengan Metamask dan ekosistem Ethereum.

REFERENCES

- [1] Felicia, Calista, S. A. Chic, M. F. Bilqisthi, and Joosten, "Tantangan dan Peluang Blockchain di Era Digital dalam Bidang Keamanan Data dan Transaksi Digital," *Journal of Comprehensive Science*, vol. 3, no. 11, pp. 5131–5147, Nov. 2024.
- [2] S. Kusumadewi, R. Kurniawan, and H. Wahyuningsih, "Implementasi Sistem Informasi Posyandu Berbasis Web Dan Android Di Desa Bimomartani," *Jurnal Pengabdian dan Pemberdayaan Masyarakat*, vol. 3, no. 2, pp. 351–359, Sep. 2019.
- [3] N. R. Abdullah, "Implementasi Teknologi Blockchain dalam Keamanan Sistem Komputer Terdistribusi," *Teknologiterkini .Org*, vol. 3, no. 7, pp. 1–11, 2023.
- [4] S. R. Aji and W. T. H. Putri, "Implementasi Teknologi Blockchain dalam Aplikasi E-Voting Berbasis Mobile," *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 14, no. 2, pp. 219–231, Nov. 2023, doi: 10.31849/digitalzone.v14i2.16682.
- [5] R. A. Fadhallah, *WAWANCARA*, 1st ed. Jakarta: UNJ PRESS, 2021.
- [6] I. Sommerville, *SOFTWARE ENGINEERING Ninth Edition*, 9th ed. Boston: Addison Wesley, 2011.