



Jurnal Teknologi Informasi dan Komunikasi

Vol: 14 No 02 2023

E-ISSN: 2477-3255

Diterima Redaksi: 26-10-2023 | Revisi: 17-11-2023 | Diterbitkan: 30-11-2023

Implementasi Teknologi Blockchain dalam Aplikasi E-Voting Berbasis Mobile

Setiawan Restu Aji¹, Wahyuningdiah Trisari Harsanti Putri²

^{1,2} Program Studi Teknik Informatika, Fakultas Ilmu Rekayasa. Universitas Paramadina
^{1,2} Jl. Raya Mabes Hankam No.Kav 9, Setu, Kec. Cipayung, Kota Jakarta Timur, DKI Jakarta
e-mail: ¹setiawan.aji@students.paramadina.ac.id, ²wahyuningdiah.trisari@paramadina.ac.id

Abstrak

Voting atau pemungutan suara adalah elemen utama dalam proses demokrasi dan pengambilan keputusan di berbagai konteks. Seiring perkembangan teknologi informasi, metode pemungutan suara tradisional telah mengalami transformasi signifikan dengan pengenalan sistem e-voting atau pemilihan elektronik. Meskipun beberapa penelitian sebelumnya telah berhasil mengembangkan aplikasi e-voting, beberapa kelemahan signifikan telah teridentifikasi, seperti masalah keamanan yang kurang memadai, risiko manipulasi, dan ketergantungan pada otoritas sentral. Teknologi blockchain memiliki sifat terdesentralisasi dan immutable. Oleh karena itu, teknologi ini menjadi solusi menarik dalam mengatasi tantangan ini. Penelitian ini bertujuan mengatasi kelemahan aplikasi e-voting sebelumnya dengan mengembangkan aplikasi e-voting Berbasis Mobile yang menggunakan teknologi blockchain. Penelitian ini juga memungkinkan partisipasi pemilih yang lebih luas melalui perangkat seluler, meningkatkan keamanan dan integritas suara. Penggunaan React Native memungkinkan aplikasi dapat diakses di berbagai platform mobile, termasuk Android dan iOS. Penelitian ini memiliki tiga aspek penting: penggunaan teknologi blockchain untuk mengatasi masalah keamanan dan integritas, pengembangan aplikasi e-voting Berbasis Mobile untuk meningkatkan aksesibilitas, dan kontribusi terhadap pengembangan aplikasi blockchain dalam berbagai konteks pemilihan.

Kata kunci: e-voting, teknologi blockchain, React Native, keamanan, transparansi, aplikasi mobile.

Implementation of Blockchain Technology in Mobile-based E-Voting Applications

Abstract

Voting is a key element in democratic processes and decision-making in various contexts. With the development of information technology, traditional voting methods have undergone a significant transformation with the introduction of e-voting systems. Although some previous research has successfully developed e-voting applications, some significant drawbacks have been identified, such as insufficient security issues, risk of manipulation, and reliance on a central authority. Blockchain technology is decentralized and immutable. Therefore, it is an attractive solution in addressing these challenges. This research aims to overcome the weaknesses of previous e-voting applications by developing a mobile-based e-voting application that uses blockchain technology. This research also enables wider voter participation through mobile devices, improving the security and integrity of votes. The use of React Native allows the application to be accessible on various mobile platforms, including Android and iOS. This research has three important aspects: the use of blockchain technology to address security and integrity issues, the development of Mobile-Based e-voting applications to improve accessibility, and the contribution to the development of blockchain application in various election context

Keywords: e-voting, blockchain technology, React Native, security, transparency, mobile application.

1. Pendahuluan

Voting atau pemungutan suara merupakan salah satu elemen utama dalam proses demokrasi dan pengambilan keputusan dalam berbagai konteks, mulai dari pemilihan umum hingga keputusan dalam organisasi-organisasi [1]. Seiring perkembangan teknologi informasi, metode pemungutan suara tradisional telah mengalami transformasi signifikan. Salah satu perkembangan penting dalam bidang ini adalah pengenalan sistem *e-voting* atau pemilihan elektronik. *E-voting* adalah penggunaan teknologi informasi dan komunikasi (TIK) untuk memfasilitasi proses pemungutan suara dalam pemilihan umum atau pemilu. Dengan *e-voting*, pemilih dapat menggunakan perangkat elektronik, seperti komputer atau ponsel pintar, untuk memberikan suara mereka.

Meskipun beberapa penelitian sebelumnya telah berhasil mengembangkan aplikasi *e-voting*, beberapa kelemahan signifikan telah teridentifikasi. Kelemahan utama termasuk kurangnya keamanan yang memadai dalam pemungutan suara elektronik, meningkatkan risiko manipulasi dan serangan siber [2], [3]. Masalah lain meliputi ketidaktransparan dalam perhitungan suara, kesulitan verifikasi hasil pemilihan, dan ketergantungan pada otoritas sentral dalam pengelolaan pemilihan. Hal tersebut bisa terjadi dikarenakan penelitian-penelitian sebelumnya dalam mengembangkan aplikasi *e-voting* menggunakan teknologi tradisional, dimana data disimpan di database sentral, sehingga data beresiko di manipulasi oleh otoritas sentral dan juga rawan serangan siber. Berdasarkan data dari Badan Siber dan Sandi Negara (BSSN) dari bulan Januari 2020 sampai April 2020 terdapat lebih dari 88.000.000 serangan siber. Pada bulan Januari terpantau 25.224.811 serangan dan kemudian pada bulan Februari terekam 29.188.645 serangan lalu kemudian pada bulan Maret terjadi 26.423.989 serangan dan sampai dengan 12 April 2020 telah tercatat 7.576.851 serangan. Puncak jumlah serangan terjadi pada tanggal 12 Maret 2020 yang mencapai 3.344.470 serangan [4].

Salah satu serangan yang berbahaya bagi sistem *e-voting* dengan teknologi tradisional adalah serangan *SQL injection*, serangan ini mampu membuat peretas mendapatkan akses ke dalam sistem dan dapat memanipulasi, merubah, dan menghapus data sistem.

Teknologi *blockchain*, dengan sifatnya yang terdesentralisasi dan *immutable*, telah menjadi solusi yang menarik dalam mengatasi tantangan ini. Teknologi ini memberikan jaminan keamanan, transparansi, dan integritas data karena sifatnya yang terdesentralisasi dan *immutable* [5]. *Blockchain* adalah basis data yang tersebar/terdesentralisasi (*decentralized database*) yang menggunakan *node* independen untuk menyimpan dan mengambil data (LaFountain, 2021). Teknologi *blockchain* memungkinkan penggabungan blok data secara berkesinambungan dalam sebuah ledger yang tersebar, di mana setiap blok memuat beragam informasi, termasuk sebuah *hash* yang berfungsi sebagai identifikasi unik dari setiap bloknnya. *Hash* melakukan identifikasi dan menautkan blok ini ke semua blok, baik blok sebelumnya dan juga blok setelahnya (Meth, 2019). Dapat disimpulkan, teknologi *blockchain* memungkinkan penggabungan atau menautkan data (*chain*) dalam buku besar terdistribusi dengan setiap blok memiliki informasi yang beragam, termasuk *hash* yang mengidentifikasi dan menghubungkan blok ke blok lainnya, baik sebelumnya maupun setelahnya.

Saat ini terdapat banyak jaringan *blockchain* yang dapat digunakan untuk membangun aplikasi terdesentralisasi, salah satunya adalah *Ethereum*. *Ethereum* merupakan sebuah jaringan *blockchain* yang terdesentralisasi, dibuat untuk menjalankan kontrak pintar (*smart contracts*) dan aplikasi terdesentralisasi (*DApps*) [3]. *Smart contract* (kontrak pintar) adalah program komputer yang berjalan di atas *platform blockchain* dan berfungsi untuk mengeksekusi dan mengotomatiskan perjanjian atau kontrak yang telah ditetapkan secara otomatis tanpa memerlukan perantara atau pihak ketiga [6].

Untuk dapat menggunakan jaringan *blockchain Ethereum* dalam lingkungan pengembangan, penulis memerlukan *tools* yang bernama *Ganache*. *Ganache* adalah perangkat lunak sumber terbuka yang memungkinkan pengembang untuk menguji, mengembangkan, dan mendeploy kontrak pintar (*smart contract*) di lingkungan pengembangan yang terisolasi [7]. Yang

dimaksud dengan lingkungan pengembangan yang terisolasi adalah suatu wadah kecil, seperti *sandbox*, yang digunakan oleh pengembang perangkat lunak untuk menguji dan mengembangkan aplikasi tanpa mempengaruhi sistem atau jaringan yang sebenarnya.

Sejumlah penelitian sebelumnya yang telah menginvestigasi implementasi teknologi *blockchain* dalam konteks *e-voting* telah membuka wawasan penting terkait dengan sistem pemilihan elektronik berbasis *blockchain*. Meskipun berbagai upaya telah dilakukan, sebagian besar penelitian sebelumnya masih mengandalkan *platform* berbasis *web* dalam pengembangan sistem *e-voting*. Berikut adalah beberapa jurnal yang telah berhasil mengimplementasikan teknologi *blockchain* ke dalam sistem *e-voting* berbasis *web*:

- Implementasi *Blockchain* dalam Aplikasi Pemilu [8]
- Implementasi *Blockchain*: Studi Kasus *e-Voting* [9]
- *Electronic Voting Using Decentralized System Based on Ethereum Blockchain* [10]

Kekurangan utama dari penelitian-penelitian tersebut adalah adalah aplikasi yang dikembangkan masih berbasis *web* yang mana memiliki kekurangan antara lain: aksesibilitas sistem yang terbatas, potensi kerentanan keamanan, serta risiko intervensi pihak ketiga dalam proses pemilihan. Sejalan dengan penelitian sebelumnya, penelitian yang berjudul "Implementasi Teknologi *Blockchain* dalam Aplikasi *E-Voting* Berbasis *Mobile*" bertujuan untuk mengatasi kelemahan tersebut. Dengan mengembangkan aplikasi berbasis *mobile*, penelitian ini menciptakan solusi peningkatan keamanan dan penggunaan yang lebih *user-friendly* sehingga memungkinkan partisipasi pemilih yang lebih luas. Solusi peningkatan keamanan yang dimaksud adalah dikarenakan teknologi *blockchain* yang digunakan dalam pengembangan aplikasi ini memiliki sifat terdesentralisasi dan *immutable* yang artinya tidak ada otoritas tunggal yang mengendalikan atau memiliki kekuasaan penuh atas jaringan dan data yang dimasukkan ke dalam *blockchain* tidak dapat diubah, hal ini memberikan keberlanjutan dan jejak riwayat yang jelas, dan dikombinasikan dengan aplikasi berbasis *mobile* yang memiliki kelebihan yaitu menggunakan metode penyandian dan enkripsi yang kuat untuk melindungi data pengguna saat berpindah antar jaringan, selain itu juga aplikasi berbasis *mobile* terhindar dari resiko serangan injeksi kode yang mana serangan ini adalah salah satu jenis serangan keamanan yang umum terjadi di aplikasi berbasis *web*. Hal ini diharapkan akan meningkatkan keamanan serta menjaga integritas suara, mendorong *e-voting* menuju keberhasilan yang lebih besar dan efisien. Dan untuk mengembangkan aplikasi *e-voting* agar dapat diakses melalui perangkat *mobile* dan juga dapat digunakan di OS *Android* maupun *iOS* (*cross-platform*), maka penulis memutuskan untuk menggunakan framework *React Native*. *React Native* merupakan sebuah *framework* pengembangan aplikasi *mobile*, yang memungkinkan pengembang membuat aplikasi *Android* dan *iOS* dengan kode sumber tunggal [11]. Peran *React Native* pada pengembangan aplikasi *e-voting* ini adalah untuk meningkatkan aksesibilitas pengguna, karena aplikasi yang telah dikembangkan dapat diakses melalui *Android* maupun *iOS*. Kombinasi teknologi *Blockchain* dan *React Native* membuka peluang baru untuk mengembangkan aplikasi *e-voting* yang aman, transparan, dan mudah digunakan oleh pengguna akhir.

Penelitian ini memiliki beberapa aspek penting. Pertama, penggunaan teknologi *blockchain* dalam pemilihan dapat mengatasi masalah keamanan, kecurangan dan manipulasi hasil pemilihan. *Blockchain* menciptakan bukti transparan dan tidak dapat diubah yang dapat diakses oleh pemilih dan pihak yang berkepentingan, seperti otoritas pemilihan dan pengamat pemilihan. Kedua, pengembangan aplikasi pemilihan Berbasis *Mobile* akan memungkinkan pemilih untuk menggunakan perangkat *mobile* mereka untuk memberikan suara. Ini adalah langkah menuju inklusivitas pemilih yang lebih besar, terutama di kalangan generasi muda yang lebih akrab dengan teknologi *mobile*. Ketiga, penelitian ini akan memberikan kontribusi terhadap pengembangan aplikasi *blockchain* yang dapat diterapkan dalam berbagai konteks pemilihan, baik dalam pemilihan politik maupun dalam pemilihan organisasi atau lembaga.

Berdasarkan latar belakang dan permasalahan diatas, berikut adalah beberapa pertanyaan yang akan menjadi acuan dari tujuan penelitian ini:

- Bagaimana mengembangkan aplikasi *e-voting* yang aman, transparan, dan terhindar dari risiko manipulasi data?
- Bagaimana merancang aplikasi *e-voting* yang dapat diakses oleh pengguna Android dan iOS?
- Bagaimana mengimplementasikan teknologi *blockchain* dalam proses voting?

Tujuan penelitian ini adalah menjawab pertanyaan-pertanyaan diatas, yaitu mengembangkan aplikasi *e-voting* yang aman, transparan, dan terhindar dari risiko manipulasi data dalam proses voting menggunakan teknologi Blockchain yang dipadukan dengan React Native, sehingga aplikasi *e-voting* yang dibangun dapat diakses oleh pengguna Android dan iOS.

2. Metode Penelitian

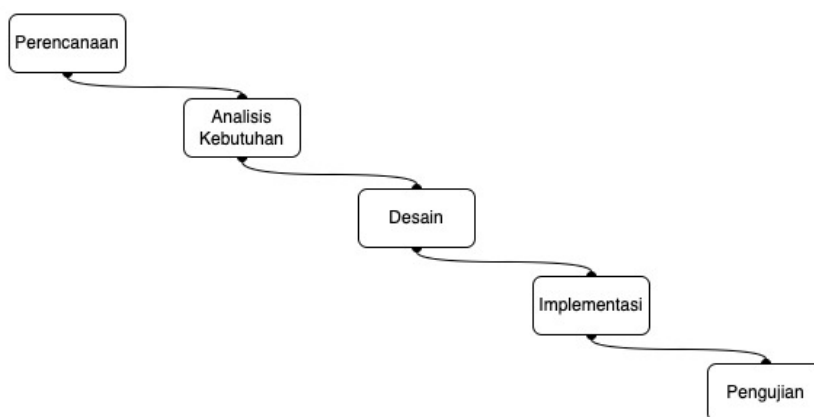
Metode yang digunakan dalam penelitian ini mencakup metode studi pustaka untuk mendapatkan pemahaman yang mendalam tentang subjek, pengembangan sistem menggunakan pendekatan metode *waterfall*.

2.1. Studi Pustaka

Metode studi pustaka adalah pendekatan penelitian yang berfokus pada pengumpulan, analisis, dan sintesis literatur terkait yang memiliki relevansi dengan topik penelitian yang sedang dijalankan [12]. Dalam konteks penelitian ini, metode studi pustaka digunakan untuk mendalami pemahaman tentang konsep dan aplikasi teknologi *blockchain*, serta untuk menggali literatur terkait *e-voting*, *React Native*, dan perpaduan keduanya.

2.2. Waterfall

Metode *Waterfall* merupakan model pengembangan perangkat lunak berbasis fase yang berurutan [13]. Kelebihan dari metode *Waterfall* adalah Model ini terdiri dari serangkaian tahap yang dilakukan secara berurutan, dimulai dari perencanaan hingga pengujian.



Gambar 1. Metode SDLC Waterfall

2.2.1. Perencanaan

Tahap perencanaan adalah awal dari seluruh proses pengembangan aplikasi. Di tahap ini, penulis merencanakan dan mendefinisikan secara rinci pengembangan aplikasi voting, termasuk tujuan, bahasa pemrograman yang akan digunakan, dan batasan-batasan aplikasi.

Tujuan dari penelitian ini adalah sesuai dengan yang telah dijelaskan pada pendahuluan, yakni menciptakan sebuah aplikasi pemilihan berbasis teknologi *Blockchain* dan *React Native* yang bertujuan untuk meningkatkan tingkat keamanan, transparansi, serta aksesibilitas dalam pelaksanaan proses pemilihan. Selanjutnya, dalam konteks pemilihan bahasa pemrograman, Penulis memilih bahasa pemrograman *Javascript* dengan *framework Express JS* untuk *backend*, bahasa pemrograman *Javascript* dengan *Framework React Native* untuk aplikasi *mobile* dan

<https://doi.org/10.31849/digitalzone.v14i2.16682>

bahasa pemrograman *Solidity* untuk pembuatan *smart contract*.

Adapun batasan-batasan aplikasi yang telah ditetapkan adalah bahwa aplikasi ini terbatas pada penggunaan di dalam jaringan *blockchain* lokal dan terisolasi dengan dukungan dari perangkat lunak bernama *Ganache*. Sehingga penulis dapat menguji dan mengembangkan aplikasi tanpa harus terhubung ke jaringan *blockchain* publik, selain itu alasan penulis menggunakan jaringan *blockchain* terisolasi adalah agar penulis memiliki kontrol penuh atas konfigurasi jaringan, termasuk jumlah node, akun pengguna, saldo awal, dan perilaku jaringan. Hal ini memungkinkan penulis untuk mensimulasikan berbagai situasi dan skenario tanpa mengkhawatirkan dampak di jaringan *Ethereum* nyata.

2.2.2. Analisis Kebutuhan

Pada tahapan ini, penulis melakukan analisis mendalam terhadap kebutuhan aplikasi *e-voting* dengan mencakup pemahaman kebutuhan pengguna, persyaratan keamanan, dan fitur-fitur yang diperlukan. Penulis kemudian mengklasifikasikan kebutuhan aplikasi *e-voting* menjadi dua kategori utama: kebutuhan fungsional dan kebutuhan non-fungsional. Pengumpulan informasi yang menjadi dasar untuk merinci kebutuhan fungsional dan non-fungsional ini dilakukan melalui studi pustaka serta observasi berbagai aplikasi *e-voting* yang ada di lingkungan sekitar.

2.2.3. Desain

Pada tahap ini, penulis akan merancang serta menjelaskan arsitektur aplikasi *e-voting*, dan juga merancang antarmuka pengguna (*UI*). penulis akan memanfaatkan *Figma* sebagai *tools* yang akan digunakan untuk merancang antarmuka pengguna (*UI*), *Figma* merupakan alat desain yang bisa dipakai di sistem operasi *Windows* maupun *Mac* untuk menciptakan prototipe aplikasi dan berbagai desain lainnya [14]. arsitektur dan antarmuka pengguna yang telah dirancang nantinya akan menjadi acuan penulis dalam membuat aplikasi *e-voting*.

2.2.4. Implementasi

Pada tahap ini penulis akan melakukan pengkodean aplikasi sesuai dengan arsitektur dan desain yang telah dibuat yaitu mencakup pembuatan *smart contract*, *backend* dan aplikasi *mobile*. *Tools* yang akan digunakan meliputi *VS Code* sebagai *code editor* dan *Ganache* sebagai jaringan *blockchain Ethereum* lokal dan terisolasi.

Penulis akan memulai dengan pembuatan sebuah *smart contract* menggunakan bahasa pemrograman *Solidity* sebagai langkah awal dalam penelitian ini. Setelah *smart contract* selesai dikodekan, langkah berikutnya adalah melakukan uji coba menggunakan perangkat lunak *Ganache* untuk memastikan bahwa kontrak tersebut berfungsi dengan baik. Setelah tahap uji coba selesai dan *smart contract* dianggap sudah siap, penulis akan melanjutkan dengan pembuatan *backend (Rest API)*. *Backend* ini akan dibangun menggunakan bahasa pemrograman *JavaScript* dan akan diperkuat dengan kerangka kerja *Express JS*, sehingga dapat memberikan kinerja yang optimal dan skalabilitas pada sistem. *Rest API* yang dibangun dalam tahap ini akan berfungsi sebagai jembatan antara aplikasi *mobile* dan *smart contract* yang telah dibuat sebelumnya. Terakhir, penulis akan fokus pada pengembangan aplikasi *mobile*, Aplikasi ini akan dikodekan menggunakan bahasa pemrograman *JavaScript* dan akan menggunakan kerangka kerja *React Native*. *Integrasi* dengan *Rest API* yang telah dibuat sebelumnya akan memungkinkan aplikasi *mobile* untuk berkomunikasi dengan *smart contract*, sehingga pengguna dapat mengakses dan berinteraksi dengan *blockchain* melalui aplikasi *mobile* dengan mudah.

2.2.5. Pengujian

Dalam tahapan pengujian, penulis akan mengimplementasikan serangkaian uji coba yang melibatkan uji fungsionalitas dan penggunaan aplikasi dengan tujuan memverifikasi bahwa setiap fitur dan fungsi yang telah diintegrasikan beroperasi sesuai yang telah ditentukan. Dalam proses ini, penulis akan mencatat hasil pengujian untuk memastikan bahwa tidak ada kegagalan sistem

dan memeriksa kemungkinan *bug* serta memastikan aplikasi bekerja dengan konsistensi dan akurasi yang diharapkan.

3. Hasil dan Pembahasan

3.1. Analisis Kebutuhan

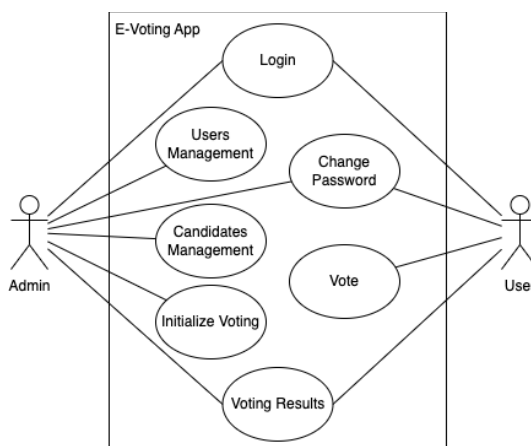
Dalam tahap analisis kebutuhan, penulis telah mengidentifikasi sumber data yang relevan dengan topik penelitian. Sumber data tersebut meliputi:

1. Jurnal ilmiah yang membahas implementasi teknologi *blockchain* dalam aplikasi *e-voting* berbasis *mobile*.
2. Artikel ilmiah yang memaparkan perkembangan terbaru dalam bidang ini.
3. Laporan penelitian terkait yang diterbitkan oleh lembaga penelitian dan organisasi terkait.
4. Sumber data primer seperti dokumen teknis dan panduan yang terkait dengan implementasi teknologi *blockchain* dalam *e-voting*.

Berdasarkan data tersebut, penulis kemudian mengklasifikasikan kebutuhan aplikasi menjadi dua kategori, yaitu *functional requirement* (kebutuhan fungsional) dan *non-functional requirement* (kebutuhan non-fungsional).

3.1.1. Functional Requirements

Dalam pengembangan aplikasi E-Voting, penulis membagi kebutuhan fungsional menjadi dua aspek, yaitu *use case diagram* yang menggambarkan fitur-fitur aplikasi dan teknologi yang akan digunakan



Gambar 2. Use case diagram aplikasi e-voting

Pada *use case diagram* diatas terlihat fitur-fitur yang nantinya akan dikembangkan oleh penulis, yaitu:

1. Terdapat 2 *role* yakni admin dan pengguna
2. Admin dan Pengguna diharuskan masuk ke aplikasi menggunakan nama pengguna (*username*) dan kata sandi (*password*) sebelum menggunakan aplikasi
3. Admin dapat membuat, mengedit, dan menghapus akun pengguna, serta mengelola hak akses pengguna.
4. Admin dapat menambahkan, mengedit, atau menghapus kandidat dalam pemilihan suara.
5. Admin juga dapat menginisialisasi pemilihan suara baru, dengan demikian sistem yang akan dibuat nantinya bisa digunakan untuk pemilihan suara lebih dari satu.
6. Pengguna hanya dapat memilih satu kandidat dalam pemilihan suara yang aktif.
7. Admin dan Pengguna juga dapat melihat hasil pemilihan suara yang telah berakhir, termasuk perolehan suara untuk setiap kandidat.

Dan Selanjutnya, dalam hal teknologi yang akan digunakan, Penulis telah mengklasifikasikannya menjadi tiga bagian, yakni:

1. *Smartcontract*

Dalam implementasi teknologi *smart contract*, penulis memilih bahasa pemrograman *Solidity*. *Solidity* adalah bahasa pemrograman yang dirancang khusus untuk pengembangan kontrak pintar (*smart contract*) di platform *blockchain Ethereum* dan dikenal karena kemampuannya dalam mengimplementasikan logika kontrak dan pemrograman berorientasi objek dalam konteks *blockchain*.

2. *Rest API (Backend)*

REST API (Representational State Transfer Application Programming Interface) adalah sekumpulan aturan dan konvensi yang digunakan untuk mengirim dan menerima data antara aplikasi dan server melalui protokol *HTTP* [15]. dalam pembuatan *Rest API* penulis menggunakan bahasa pemrograman *Javascript* dengan *framework ExpressJS*

3. *Aplikasi Mobile (Frontend)*

Sedangkan untuk aplikasi *mobile* penulis menggunakan bahasa pemrograman *Javascript* dengan *framework React Native*.

3.1.2. Non-Functional Requirements

Dalam rangka mencapai kesuksesan mengimplementasikan teknologi *blockchain* ke aplikasi *e-voting* Berbasis *Mobile*, penulis telah menganalisis sejumlah kebutuhan non-fungsional yang akan menjadi fokus dalam penelitian ini. Berikut adalah daftar kebutuhan non-fungsional untuk aplikasi *e-voting*:

4. Keamanan Data

Data hasil pemilihan harus disimpan di dalam jaringan *blockchain*, yang menawarkan tingkat keamanan dan integritas data yang tinggi

5. Transparansi

Data pemilihan harus tersedia untuk inspeksi publik secara realtime tanpa mengungkapkan identitas pemilih

6. Ketersediaan

Aplikasi harus tersedia sepanjang waktu, khususnya selama periode pemilihan

7. Aksesibilitas

Aplikasi harus dapat diakses dan digunakan melalui berbagai platform *mobile*, termasuk *Android* dan *iOS*

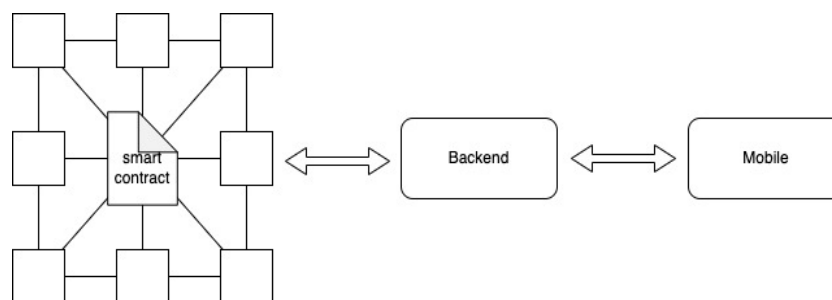
8. Performa

Aplikasi harus memberikan waktu respons yang cepat kepada pemilih, terutama saat melakukan pemilihan dan melihat hasil.

3.2. Desain

Seperti yang telah dijelaskan pada metode penelitian, pada tahapan desain penulis telah merancang arsitektur aplikasi dan juga antarmuka aplikasi *e-voting*.

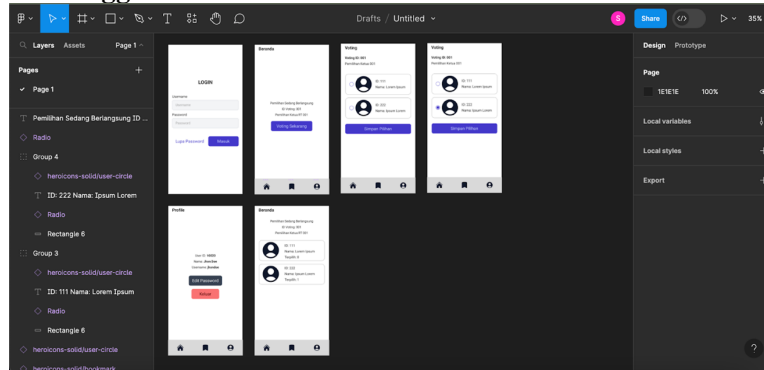
3.2.1. Arsitektur Aplikasi



Gambar 3. Arsitektur aplikasi e-voting

Berdasarkan arsitektur diatas dapat dilihat bahwa aplikasi *e-voting* yang dibuat tidak langsung terhubung dengan *smart contract* yang ada di jaringan *blockchain*, melainkan akan dihubungkan melalui *backend*, peran *backend* pada arsitektur ini sangatlah penting, karena selain penghubung aplikasi mobile dengan *smart contract*, didalam *backend* juga nantinya akan ada beberapa tambahan fitur keamanan, hal ini agar memastikan bahwa sistem yang dibuat benar-benar aman.

3.2.2. Antarmuka Pengguna



Gambar 4. Antarmuka pengguna

Pada gambar diatas penulis telah menggunakan *tools Figma* untuk merancang antarmuka pengguna (*UI*) aplikasi. Desain yang telah dibuat mencakup tampilan dari aplikasi *e-voting* yang akan dikembangkan, beberapa diantaranya adalah tampilan *login*, *profile*, kandidat, dan hasil pemilihan.

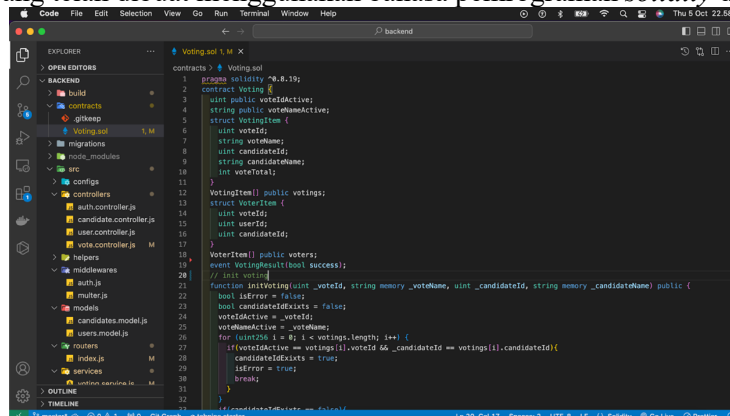
3.3. Implementasi

Pada tahapan implementasi ini, dijelaskan secara rinci proses implementasi teknologi *blockchain* dalam aplikasi *e-voting* berbasis *mobile*. Tahap ini terbagi menjadi beberapa bagian utama agar pengembangan dapat berlangsung secara sistematis sesuai dengan tujuan.

3.3.1. Pembuatan Smart Contract

Pada proses pembuatan *smart contract* untuk aplikasi *e-voting*, serangkaian langkah penting telah dijalankan untuk menjamin integritas serta keamanan proses pemilihan elektronik. Proses dimulai dengan perancangan kontrak pintar yang memuat pemilihan struktur data yang sesuai guna menyimpan data pemilih, kandidat, serta hasil pemilihan.

Setelah desain kontrak pintar selesai, langkah berikutnya adalah mengkodekan kontrak menggunakan bahasa *Solidity*. Pengkodean melibatkan implementasi mekanisme pemungutan suara, verifikasi pemilih, dan penghitungan suara, dengan penekanan pada praktik terbaik keamanan *Solidity* untuk menghindari potensi kerentanan. Berikut adalah *screenshot* dari kode *smart contract* yang telah dibuat menggunakan bahasa pemrograman *solidity* dan *tools VS Code*:



Gambar 5. Screenshot kode smart contract

<https://doi.org/10.31849/digitalzone.v14i2.16682>

Setelah selesai mengkodekan, *smart contract* yang baru dibuat diuji menggunakan *Ganache*, suatu lingkungan pengujian lokal yang terisolasi. Uji coba ini dilakukan untuk memastikan bahwa *smart contract* beroperasi sesuai yang diharapkan, mematuhi aturan pemilihan yang telah ditetapkan, dan terhindar dari potensi kesalahan yang dapat mengancam keamanan dan integritas data.

3.3.2. Pembuatan Backend

Setelah *Smart Contract* telah selesai dibuat dan juga telah diuji menggunakan *Ganache*, langkah selanjutnya adalah pembuatan *Rest API* atau *backend* menggunakan bahasa pemrograman *Javascript* dengan *framework ExpressJS*. *Backend* ini nantinya akan digunakan sebagai *gateway* dan penghubung antara aplikasi *mobile* dengan *smart contract* yang telah dibuat.

Proses dimulai dengan merancang arsitektur *backend* secara mendalam. Ini mencakup pemilihan komponen, struktur basis data, dan perencanaan rute *API*. Tujuannya adalah memastikan bahwa *backend* dapat diakses oleh aplikasi *mobile* dan juga dapat mengakses data dari *smart contract* dengan efisien.

Setelah desain *backend* selesai, langkah selanjutnya adalah mengimplementasikan *backend* menggunakan *ExpressJS*. Memastikan bahwa *backend* dapat berinteraksi dengan *smart contract* yang ada melalui koneksi *blockchain* yang sesuai.

Berikut adalah *screenshot* kode dari *backend* yang yang dihubungkan dengan *smart contract*:

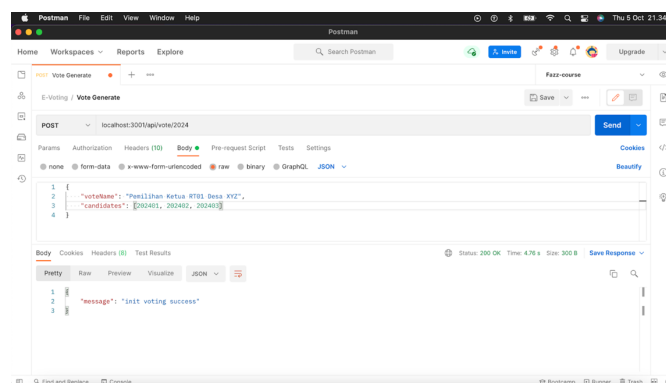
```

1 const Web3 = require('web3')
2 const CONTRACT_ABI = require('../build/contracts/Voting.json')
3 const { CONTRACT_ADDRESS, WEBSOCKET_URL, SENDER_ADDRESS } = require('../helpers/env')
4 const web3 = new Web3(WEBSOCKET_URL)
5 const contractAddress = "0x0a569d188bf68c2758ba146d4e1378aa85945"
6 const senderAddress = SENDER_ADDRESS
7 const contract = new web3.eth.Contract(CONTRACT_ABI, contractAddress)
8 module.exports = {
9   createCandidate: async (_voteId, _voteName, _candidateId, _candidateName) => {
10     try {
11       const encodedData = await contract.methods.initVoting(_voteId, _voteName, _candidateId, _candidateName).encodeABI()
12       const estimateGas = await contract.methods.initVoting(_voteId, _voteName, _candidateId, _candidateName).estimateGas({ from: senderAddress, data: encodedData })
13       const transaction = await contract.methods.initVoting(_voteId, _voteName, _candidateId, _candidateName).send({ from: senderAddress, gas: estimateGas })
14     } catch (error) {
15       console.error('Error: ', error)
16     }
17   }
18 }
19 const logs = await contract.getPastEvents('VotingResult', { fromBlock: transaction.blockNumber, toBlock: transaction.blockNumber })
20 const result = logs[0].returnValues.success
21 return result
22 } catch (error) {
23   console.error('Error: ', error)
24 }
25 }
26 }
27 }
28 }
29 }
30 }
31 }
32 }

```

Gambar 6. Screenshot kode backend

Pada Gambar diatas terlihat bahwa terdapat alamat dari *smart contract* (*contractAddress*), alamat *smart contract* ini digunakan untuk menghubungkan *backend* dengan *smart contract* yang sudah ada, dan juga terdapat kode fungsionalitas dari aplikasi seperti inisialisasi voting, melakukan vote, mendapatkan hasil voting, dll.



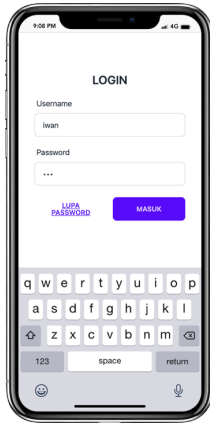
Gambar 7. Screenshot pengujian backend menggunakan postman

Setelah *backend* selesai diimplementasikan dan diintegrasikan dengan *smart contract*, langkah selanjutnya adalah melakukan serangkaian pengujian untuk memastikan bahwa semua fungsi beroperasi sesuai dengan yang diharapkan. Pengujian mencakup pengujian fungsionalitas, keamanan, dan kinerja untuk memastikan bahwa *backend* siap digunakan dalam pemilihan elektronik. Untuk pengujian *backend* penulis menggunakan *tools* yang bernama *Postman*.

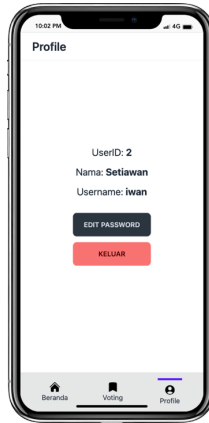
Postman adalah sebuah platform kolaborasi yang digunakan oleh pengembang perangkat lunak untuk membangun, menguji, dan mendokumentasikan *API (Application Programming Interface)* [16]. Berikut adalah *screenshot* dari pengujian *Backend* menggunakan *Postman* untuk menginisialisasi voting

3.3.3. Pembuatan Aplikasi Mobile

Setelah pembuatan *backend* selesai langkah selanjutnya adalah pembuatan aplikasi *mobile* menggunakan bahasa pemrograman *Javascript* dengan *framework React Native*. Proses dimulai dengan merancang antarmuka pengguna (*UI*) aplikasi *mobile* yang sudah dibuat menggunakan *tools Figma* pada tahapan Desain. Setelah desain *UI* selesai dibuat, langkah selanjutnya adalah pengembangan fungsionalitas inti aplikasi *mobile* dengan mengintegrasikan aplikasi *mobile* dengan *Backend* yang telah dibuat. Ini mencakup pengembangan sistem untuk *login*, menampilkan data kandidat, melakukan voting, dan melihat hasil pemilihan yang diambil dari *backend* dan *smart contract*. Fitur-fitur keamanan diterapkan untuk melindungi data dan integritas pemilihan. Berikut adalah *screenshot* aplikasi yang telah dibuat



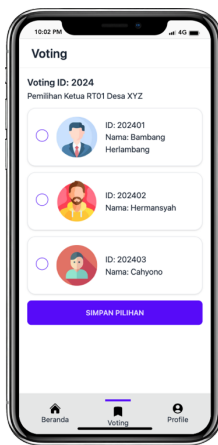
Gambar 8. Tampilan login



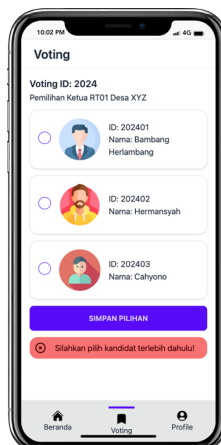
Gambar 9. Tampilan profile



Gambar 10. Tampilan beranda



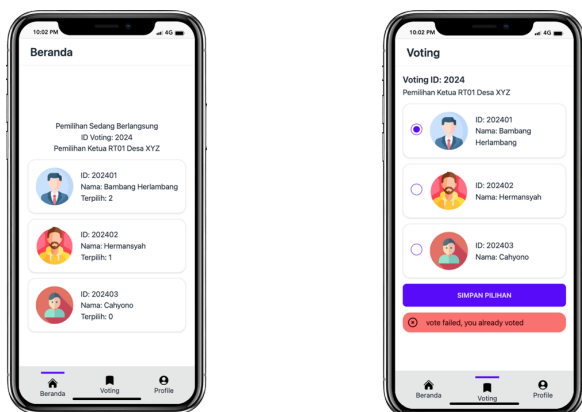
Gambar 11. Tampilan voting



Gambar 13. Tampilan ketika user melakukan voting

<https://doi.org/10.31849/digitalzone.v14i2.16682>

Gambar 12. Tampilan ketika user melakukan voting



Gambar 14. Tampilan hasil voting

Gambar 15. Tampilan ketika user melakukan voting lebih dari satu kali

3.4. Pengujian

Aplikasi *mobile* yang telah selesai dibuat kemudian diuji guna memverifikasi fungsionalitasnya. Seluruh skenario pengujian diterapkan pada aplikasi tersebut, dan hasilnya menunjukkan bahwa aplikasi *mobile* telah berhasil melewati seluruh pengujian yang telah dilakukan.

dan berikut adalah catatan hasil pengujian yang telah dilakukan oleh penulis

Table 1. Hasil pengujian aplikasi

Test ID	Deskripsi	Ekspektasi	Hasil	Keterangan
1.	pengguna dan admin dapat masuk menggunakan username dan password yang valid	Masuk ke aplikasi	Berhasil masuk ke aplikasi	Success
2.	admin dapat membuat akun pengguna	Data pengguna baru ditambahkan	Berhasil menambahkan pengguna baru	Success
3.	admin dapat mengedit informasi akun pengguna	Data pengguna berubah	Berhasil mengedit informasi pengguna	Success
4.	admin dapat menghapus akun pengguna jika voting sudah tidak aktif	Data pengguna terhapus	Berhasil menghapus data pengguna	Success
5.	admin dapat membuat kandidat baru	Data kandidat baru ditambahkan	Berhasil menambahkan kandidat baru	Success

6.	admin dapat mengedit informasi kandidat	Data kandidat berubah	Berhasil mengedit informasi kandidat	Success
7.	admin dapat menghapus kandidat jika voting sudah tidak aktif	Data kandidat terhapus	Berhasil menghapus data kandidat	Success
8.	admin dapat menginisialisasi pemilihan suara baru	Voting baru ditambahkan dan menonaktifkan voting sebelumnya	Berhasil menambahkan voting baru dan menonaktifkan voting sebelumnya	Success
9.	pengguna dapat melihat pemilihan suara yang aktif beserta kandidatnya	Menampilkan voting yang aktif beserta kandidatnya	Berhasil menampilkan data voting yang aktif beserta data kandidat	Success
10.	pengguna hanya dibolehkan memilih satu kandidat	Hanya dapat memilih satu kandidat saat voting	Berhasil memilih hanya satu kandidat	Success
11.	Pengguna hanya dibolehkan melakukan voting sekali	Hanya dapat melakukan voting sekali	Berhasil melakukan voting sekali	Success
12.	admin dan pengguna dapat melihat hasil pemilihan suara	Menampilkan hasil voting beserta score masing-masing kandidat	Berhasil menampilkan hasil voting beserta score masing-masing kandidat	Success

3.5. Perbandingan dengan penelitian terdahulu

Meskipun penelitian ini dan penelitian-penelitian sebelumnya telah berhasil menciptakan aplikasi *e-voting* berbasis *blockchain* yang terdesentralisasi, immutable dan memiliki tingkat keamanan yang tinggi, namun hasil penelitian-penelitian sebelumnya masih menggunakan *website* sebagai basis aplikasi *blockchain*[8][9][10]. Dibandingkan dengan penelitian-penelitian sebelumnya, penelitian ini menggunakan perangkat *mobile* sebagai basis aplikasi *blockchain*, kelebihan dari aplikasi *mobile* adalah aksesibilitas dan kemudahan penggunaan. Aplikasi *mobile* memiliki kelebihan yaitu terhindar dari resiko serangan injeksi kode yang merupakan salah satu jenis serangan keamanan yang umum terjadi di aplikasi berbasis *web*, sehingga pengimplementasian *blockchain* pada aplikasi berbasis *mobile* lebih baik dibandingkan dengan aplikasi *blockchain* berbasis *web*.

4. Kesimpulan

Dalam penelitian ini, kami berhasil mengembangkan aplikasi *e-voting* berbasis *mobile cross-platform* dengan menggunakan teknologi *Blockchain* dan *React Native*. Tujuan utama kami adalah meningkatkan keamanan, transparansi, efisiensi, dan aksesibilitas dalam proses pemilihan. Penelitian ini mengadopsi pendekatan pengembangan aplikasi *e-voting* berbasis *mobile*, yang menjanjikan tingkat keamanan dan aksesibilitas lebih tinggi dibandingkan dengan penelitian sebelumnya yang masih menggunakan pengembangan berbasis *web*. Kami menerapkan pendekatan *Waterfall* dari perencanaan hingga pengujian. Pengujian aplikasi mencakup uji fungsionalitas dan penggunaan aplikasi untuk memastikan kesesuaian dengan spesifikasi yang

ditetapkan. Penelitian ini merupakan langkah awal yang menjanjikan dalam menggabungkan teknologi blockchain dengan aplikasi e-voting berbasis mobile, menciptakan aplikasi voting yang aman dan efisien. Hasil penelitian menunjukkan potensi besar aplikasi ini dalam meningkatkan keamanan dan integritas pemilihan. Dengan pemeliharaan yang berkelanjutan, penerapan regulasi yang tepat, dan kerjasama yang kuat dengan pemangku kepentingan, aplikasi ini dapat menjadi solusi yang berharga untuk pemilihan masa depan.

Daftar Pustaka

- [1] L. Hardjaloka And V. M. Simarmata, "E-Voting: Kebutuhan Vs. Kesiapan (Menyongsong) E-Demokrasi." <https://doi.org/10.31078/jk847>.
- [2] Beni Willia Saputra, "Tindak Lanjut Terhadap Penerapan Elektronik Voting Dalam Pelaksanaan Pemilihan Kepala Daerah Berdasarkan Peraturan Perundang-Undangan," *Journal Of Constitutional Law*, Vol. 1, 2021, Doi: <https://doi.org/10.22437/Limbago.V1i2.14377>.
- [3] F. Diny Hermawati And M. Tahir, "Keamanan E-Voting Di Indonesia Melalui Pemanfaatan Kriptografi Pada Sistem Aes (Advance Encryption Standard)," *Pendidikan Informatika*, 2023. Doi: <https://doi.org/10.55606/jtmei.V2i2.1625>.
- [4] B. H. Dan K. Bagian Komunikasi Publik, "Rekap Serangan Siber (Januari – April 2020)," 2020.
- [5] M. Oka Augusta, C. Putriana Oktaviandi Syeira, A. Hadiapurwa, And K. Kunci, "Penggunaan Teknologi Blockchain Dalam Bidang Pendidikan," Vol. 437, No. 2, 2021, Doi: <https://doi.org/10.35568/Produktif.V5i2.1259>.
- [6] Effrida Ayni Fikri And Teddy Anggoro, "Penggunaan Smart Contract Pada Teknologi Blockchain Untuk Transaksi Jual Beli Benda," *Jurnal Ilmu Sosial Dan Pendidikan (Jisip)*, 2022, Doi: <http://dx.doi.org/10.58258/Jisip.V6i3.3301>.
- [7] J. P. Hukum, M. S. Aris, D. Elvia, K. Putri, G. A. Putra, And X. Nugraha, "The Blockchain-Based On E-Voting In The Local Volume 10 No. 1 Januari-April 2023 Elections System: An Effort To Realize E-Democracy The Blockchain-Based On E-Voting In The Local Elections System: An Effort To Realize E-Democracy," 2023. Doi: <http://dx.doi.org/10.26532/Jph.V10i1.22811>.
- [8] Yulfitno Wingga Pratama, "Implementasi Blockchain Dalam Aplikasi Pemilu," *Jurnal Vocational Teknik Elektronika Dan Informatika*, Sep. 2021, Doi: <https://doi.org/10.24036/Voteteknika.V9i3.113787>.
- [9] S. Damai, K. Hu, H. Novianus Palit, And A. Handojo, "Implementasi Blockchain: Studi Kasus E-Voting."
- [10] F. Jri Fadli, S. Jatmiko, And M. Lamsani, "Electronic Voting Using Decentralized System Based On Ethereum Blockchain," *Jurnal Ilmiah Komputasi*, Vol. 19, 2020, <http://dx.doi.org/10.32409/Jikstik.19.1.2718>.
- [11] A. Husein Malahella And I. Arwani, "Pemanfaatan Framework React Native Dalam Pengembangan Aplikasi Pemesanan Minuman Kopi Pada Kedai Bycoffee," 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] M. Nina Adlini, A. Hanifa Dinda, S. Yulinda, O. Chotimah, And S. Julia Merliyana, "Metode Penelitian Kualitatif Studi Pustaka," 2022. Doi: <https://doi.org/10.33487/Edumaspul.V6i1.3394>.
- [13] W. Steven Dharmawan *Et Al.*, "Penerapan Metode Sdlc Waterfall Dalam Perancangan Sistem Informasi Administrasi Keuangan Berbasis Desktop," Vol. Vi, No. 2, 2018, Doi: <https://doi.org/10.31294/Jki.V6i2.5733>.
- [14] A. J. Publikasi *Et Al.*, "Mengenal Aplikasi Figma Untuk Membuat Content Menjadi Lebih Interaktif Di Era Society 5.0," Vol. 1, No. 6, Pp. 552–555, 2023, [Online]. Available: <https://jurnal.portalpublikasi.id/index.php/ajp/index>
- [15] I. A. Kaniya *Et Al.*, "Perancangan Dan Implementasi Restful Api Pada Sistem Informasi Manajemen Dosen Universitas Udayana," 2022. Doi: <https://doi.org/10.24843/Spektrum.2022.V09.I03.P3>.
- [16] W. Galindra Wardhana, I. Arwani, And B. Rahayudi, "Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)," 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>