

Implementation of Blockchain Technology in E-Voting Systems

Penerapan Teknologi *Blockchain* Pada Sistem *E-Voting*

Gideon L. Saroinsong, Alwin Sambul, Sherwin R. U. A. Sompie

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails : deonsaroinsong.ds@gmail.com, asambul@unsrat.ac.id, aldo@unsrat.ac.id

Received: 21 July 2024; revised: 27 October 2024; accepted: 10 December 2024

Abstract — *In the current digital era, many technologies have been implemented in various systems, including voting. Conventional voting methods often face several issues such as lack of transparency, paper ballot damage, and difficulties in supervision. E-voting technology emerges as a solution to address these problems but still faces challenges related to security and data integrity. This research designs a voting system using local blockchain technology implemented with the Hardhat network. The study shows that the blockchain-based e-voting system with smart contracts provides enhanced security and transparency in the voting process. The voting data can be verified by all users to ensure that votes remain unchanged and are securely stored in the blockchain, thereby increasing trust and accountability in the election system. Thus, blockchain technology offers an innovative and effective solution to the existing problems in conventional voting systems. Additionally, this system offers higher flexibility and accessibility compared to traditional methods. The results of this research are expected to serve as a reference for the implementation of e-voting systems in the future.*

Key words— *E-Voting; Blockchain; Security; Transparency.*

Abstrak — Dalam era digital saat ini, banyak teknologi telah diimplementasikan dalam berbagai sistem, termasuk voting. Metode pemungutan suara konvensional sering menghadapi berbagai masalah seperti kurangnya transparansi, kerusakan kertas suara, dan kesulitan dalam pengawasan. Teknologi *e-voting* hadir sebagai solusi untuk mengatasi masalah-masalah tersebut, namun masih dihadapkan pada tantangan keamanan dan integritas data. Pada penelitian ini dirancang suatu sistem pemungutan suara dengan menggunakan teknologi *blockchain* lokal dengan menggunakan jaringan *Hardhat*. Penelitian ini menunjukkan bahwa sistem *e-voting* yang dibuat berbasis *blockchain* dengan *smart contract* memberikan hasil keamanan dan transparansi pada proses pemilihan. Data pemilihan dapat diperiksa oleh semua pengguna untuk memastikan bahwa suara tetap terjaga dan suara sudah tersimpan dalam *blockchain*, sehingga meningkatkan kepercayaan dan akuntabilitas dalam sistem pemilihan. Dengan demikian, teknologi *blockchain* menawarkan solusi yang inovatif dan efektif dalam mengatasi permasalahan yang ada dalam sistem pemungutan suara konvensional. Selain itu, sistem ini memberikan fleksibilitas bisa diakses dimanapun dibandingkan metode konvensional. Hasil penelitian ini diharapkan dapat menjadi acuan untuk implementasi sistem *e-voting* di masa depan.

Kata kunci — *E-Voting; Blockchain; Keamanan; Transparan.*

I. PENDAHULUAN

Dalam era digital saat ini, teknologi semakin menjadi bagian penting dari berbagai aspek kehidupan manusia, termasuk dalam proses pengambilan keputusan melalui voting atau pemungutan suara. Voting adalah metode yang digunakan untuk mengambil keputusan dalam berbagai konteks, mulai dari pemilihan umum hingga pemilihan organisasi kecil. Namun, metode konvensional pemungutan suara sering kali menghadapi berbagai masalah, seperti transparansi yang kurang, kertas suara yang rusak, kekurangan saksi, hingga permasalahan dalam perhitungan suara.

Pada Pemilu 2024, Bawaslu menemukan sejumlah masalah dalam pengadaan dan distribusi logistik Pemilu. Beberapa masalah termasuk kerusakan kotak suara di 177 kabupaten/kota, bilik suara rusak di 61 kabupaten/kota, dan tinta rusak di 124 kabupaten/kota. Selain itu, ada pengiriman yang salah tujuan di 10 kabupaten/kota. Tahap kedua pengawasan menemukan surat suara rusak di 127 kabupaten/kota dan kekurangan jumlah di 61 kabupaten/kota. Bawaslu juga mencatat masalah pengawasan logistik di luar negeri, seperti kekurangan dan kerusakan surat suara di berbagai Panitia Pemilihan Luar Negeri (PPLN).

Masalah-masalah tersebut menunjukkan perlunya sebuah sistem yang lebih transparan dan aman untuk mengelola proses pemungutan suara. Teknologi *Blockchain*, dengan kemampuannya untuk mencatat setiap transaksi secara transparan dan aman, menawarkan solusi yang potensial untuk meningkatkan integritas data suara yang ada dalam sistem *e-voting*. *Blockchain* adalah teknologi buku besar terdistribusi yang mencatat setiap transaksi dalam bentuk blok yang saling terhubung dan tidak dapat diubah. Setiap data yang tercatat dalam *Blockchain* dapat diverifikasi dan dilacak tanpa mengungkapkan identitas pemilih, sehingga menjaga privasi dan anonimitas.

Penelitian ini bertujuan untuk mengkaji penerapan teknologi *Blockchain* dalam sistem *e-voting*, mengidentifikasi tantangan yang ada, dan menawarkan solusi yang dapat mengatasi masalah tersebut. Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam mengembangkan sistem pemilihan yang lebih adil, transparan, dan aman, serta mendorong adopsi teknologi *Blockchain* dalam berbagai aplikasi lainnya.

A. Penelitian Terkait

Penelitian terkait tentang penelitian sebelumnya berkaitan dengan penerapan teknologi *blockchain* pada sistem *e-voting* yang menjadikan bagai bahan acuan dalam pelaksanaan penelitian ini, penelitiannya sebagai berikut.

The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. Penelitian terhadap sistem pemungutan suara online iVote yang digunakan dalam pemilihan negara bagian New South Wales pada tahun 2015 menunjukkan adanya kerentanan serius yang dapat dimanfaatkan untuk memanipulasi suara, melanggar privasi pemilih, dan merusak mekanisme verifikasi. menegaskan pentingnya mencari solusi yang lebih aman dan tahan terhadap serangan yang dapat diatasi[1].

Implementasi *E-Voting* Pemilu Raya Mahasiswa Universitas Primakara Berbasis Web Studi Kasus Pada Universitas Primakara) Implementasi *e-voting* dilakukan dengan menggunakan framework Laravel untuk pembuatan program aplikasi akademik berbasis web, tahapan-tahapan yang dilakukan meliputi analisis kebutuhan, desain, pengumpulan data, analisis data, perancangan, dan pembuatan kesimpulan penelitian. Dalam membantu penulis dalam melaksanakan pembuatan tahapan-tahapan pembuatan sistem *e-voting*[2].

Sistem *e-voting* dalam pemilihan kepala daerah 2020 saat pandemi covid-19: Perbandingan Indonesia, Australia dan Brazil. Penelitian ini membandingkan sistem *e-voting* yang digunakan dalam pemilihan kepala daerah di Indonesia, Australia, dan Brazil selama pandemi Covid-19. Hasil penelitian menunjukkan bahwa sistem *e-voting* mampu meningkatkan partisipasi publik serta memberikan keefektifan dan efisiensi dalam proses pemilihan. Namun, penerapan sistem ini masih menghadapi tantangan terkait keamanan dari serangan hacker dengan sistem yang dibuat bertujuan untuk meningkatkan partisipasi dan efisiensi pemilu di Indonesia, menyimpulkan bahwa *e-voting* menjadi solusi yang bisa meningkatkan partisipasi voter[3].

Penerapan Asas Pemilu Terhadap Electronic Voting (*E-Voting*) Pada Pemilu Tahun 2024. Penelitian ini membahas efektivitas sistem *e-voting* dalam pemilihan umum yang telah memenuhi prinsip-prinsip pemilu yang jujur dan adil (Luber Jurdil). Penelitian ini juga mengusulkan pembentukan regulasi khusus dan eksplisit, baik dalam bentuk undang-undang maupun Perppu, untuk mendukung penyelenggaraan *e-voting* di masa mendatang lewat penelitian tersebut dengan sistem yang akan dibuat menunjukkan bahwa indonesia siap untuk melaksanakan pemilu secara *e-voting*[4].

Implementasi *Blockchain*: Studi Kasus *e-Voting* Penelitian ini mengkaji implementasi sistem *e-voting* berbasis *blockchain* dengan menggunakan Multichain sebagai alat untuk membangun dan menjalankan aplikasi *blockchain*. Studi kasus ini menekankan manfaat *blockchain* dalam menyediakan transparansi dan integritas data suara, yang dapat diverifikasi oleh semua pihak yang terlibat dalam proses pemilihan sistem yang dibuat juga menggunakan *blockchain* untuk memastikan suara yang transparan dan terintegritas[5].

Secure Digital Voting System based on *Blockchain* Technology. Penelitian ini menekankan pentingnya menjaga anonimitas dan integritas suara serta memastikan keterjangkauan dan kemudahan penggunaan bagi pemilih.

Penelitian ini menunjukkan bahwa dengan teknologi *blockchain* bisa menjaga anonimitas dan integritas dari suara voter yang memilih candidate[6].

Smart Contract Blockchain pada *E-Voting* Penelitian ini fokus dalam melakukan pengujian keamanan menggunakan *smart contract blockchain* pada *e-voting* dengan menggunakan bahasa *Solidity*, dengan penelitian tersebut bisa menunjukkan tentang bagaimana *smart contract* bekerja dalam pengujian dan keamanan sehingga penelitian tersebut bisa membantu dalam proses pengembangan *smart contract*[7].

Aplikasi Voting Online dengan Menggunakan Teknologi *Blockchain*. *Blockchain* bertugas untuk mengumpulkan suara setiap daerah. Setelah *voting* selesai semua hasil *Blockchain* dikumpulkan menjadi satu. Hasil pada *blockchain* tidak dapat diganti karena masih dalam chipertext Melibatkan deskripsi menggunakan kunci privat dari stakeholder setelah pengumpulan hasil dari *blockchain*[8].

B. E-Voting

E-Voting, atau pemungutan suara elektronik, merujuk pada metode pemilihan umum yang memanfaatkan alat elektronik untuk mempermudah proses pemungutan suara. Kelebihan utama dari *E-Voting* meliputi proses pemilihan yang cepat, perhitungan suara yang instan, dan tampilan real-time hasil suara pada aplikasi terkait. Dengan demikian, setiap suara yang masuk dapat dengan segera terlihat, meningkatkan transparansi dan aksesibilitas informasi. Selain itu, implementasi *E-Voting* bertujuan meminimalisir kelemahan yang mungkin terdapat pada metode pemungutan suara konvensional[9].

C. Blockchain

Blockchain adalah suatu teknologi yang memungkinkan pembuatan dan penyimpanan catatan transaksi dalam bentuk rantai blok yang aman dan terdesentralisasi, Rantai blok ini terdiri dari blok-blok yang saling terhubung, dan setiap blok menyimpan serangkaian [10].

Setiap blok berisi data, *Hash* unik, dan *Hash* dari blok sebelumnya, dengan *Hash* Sebuah kode unik yang dihasilkan dari data dalam blok, perubahan kecil pada data akan menghasilkan *Hash* yang sangat berbeda, setiap blok terhubung ke blok sebelumnya melalui *Hash*, dan membentuk rantai yang aman.

D. Ethereum

Ethereum adalah sebuah platform *blockchain* terdesentralisasi yang diperkenalkan pada tahun 2015 oleh Vitalik Buterin. Platform ini memungkinkan pengembangan dan pelaksanaan *smart contract*, yaitu kode komputer yang otomatis mengeksekusi perjanjian atau protokol. Ether (ETH) adalah mata uang kripto yang digunakan di dalam ekosistem Ethereum untuk membayar biaya transaksi dan pelaksanaan *smart contract*. Ethereum merupakan jaringan terdesentralisasi, di mana setiap node di jaringan memiliki salinan lengkap dari *blockchain* Ethereum[11].

E. Smart Contract

Smart contract adalah program komputer yang dirancang untuk secara otomatis mengeksekusi, menegosiasikan, atau menegaskan suatu kesepakatan atau kontrak tanpa memerlukan pihak ketiga. Konsep ini pertama kali diperkenalkan oleh Nick

Szabo pada tahun 1994, dan *smart contract* menggunakan teknologi *blockchain* untuk memastikan keamanan, transparansi, dan tidak dapat diubahnya hasil eksekusi, bahasa yang digunakan adalah *Solidity* merupakan bahasa pemrograman yang dirancang khusus untuk menulis *smart contract* di atas platform Ethereum.

Penggunaan teknologi *blockchain* dalam *smart contract* memastikan keamanan karena informasi yang disimpan di dalamnya dienkripsi dan tersebar di seluruh jaringan *blockchain*, sehingga sulit untuk dimanipulasi atau diakses oleh pihak yang tidak berwenang[12].

F. Hash

Hash adalah fungsi yang mengambil input data dengan ukuran berapa pun dan menghasilkan output dengan ukuran tetap yang unik, yang disebut *Hash value* atau *Hash code*. *Hash* digunakan dalam berbagai aplikasi seperti penyimpanan data, pencarian data, tanda tangan digital, dan verifikasi integritas data. Dalam konteks *blockchain*, *Hash* digunakan untuk mengidentifikasi blok, memastikan integritas data, dan mengamankan transaksi[13].

G. Hardhat

Hardhat adalah lingkungan pengembangan (development environment) untuk Ethereum yang membantu developer dalam membuat, menguji, dan menerapkan *decentralized application* dan *smart contract*. *Hardhat* menawarkan berbagai fitur dan alat yang mempermudah proses pengembangan, seperti pengujian (testing) unit dan integrasi untuk memastikan *smart contract* bekerja dengan benar sebelum diterapkan di jaringan utama.

Hardhat menyediakan alat debugging, termasuk *Hardhat Network*, jaringan *blockchain* lokal yang cepat dan mudah digunakan untuk simulasi dan pengujian. Fitur penyebaran (deployment) juga disediakan oleh *Hardhat*, memungkinkan developer untuk menerapkan *smart contract* mereka ke berbagai jaringan Ethereum dengan[14].

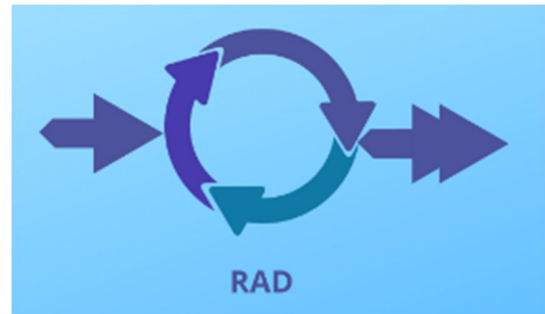
II. METODE

A. Waktu dan Tempat Penelitian

Penelitian ini direncanakan akan dilaksanakan selama 4 bulan, dimulai dari bulan February 2024 hingga Mei 2024, Tempat penelitian dilaksanakan di Rumah dan Lingkungan Tempat Pemilihan Umum yaitu di Desa Sapa, Kec. Tenga, Kab. Minahasa Selatan 2024, semuanya sudah termasuk dari studi pustaka sampai pembuatan aplikasi dan penyusunan laporan.

B. Metode Penelitian

Dalam mendukung pengembangan sistem penulis memerlukan metode penelitian yang menentukan kualitas dari sistem aplikasi yang akan dibangun. Dalam penelitian ini, metode penelitian yang akan diterapkan dalam sistem ini adalah metode Rapid Application Development (RAD) merupakan metode pengembangan perangkat lunak yang menekankan pada kecepatan dan fleksibilitas[15].



Gambar 1. Metode Rapid Application Development

1) Perencanaan Kebutuhan

Merupakan proses penulis mengidentifikasi dan menentukan kebutuhan dan persyaratan dari sistem. pada bagian ini seluruh penelitian lewat wawancara selama proses pemilihan umum di Desa Sapa dan studi pustaka yang dilakukan sendiri di Rumah, dan data yang akan didapatkan merupakan perbandingan antara sistem secara tradisional dan berbasis *blockchain*.

2) Workshop Design

Tahap Ini adalah tahap untuk merancang dan memperbaiki yang bisa dikatakan sebagai workshop. Dimana penganalisis bekerja membangun dan membuat perancangan representasi desain kerja pengguna. Pada tahap ini penulis menyusun dasar dalam pembuatan aplikasi e-voting dengan menggambarkan desain awal dalam memberikan gambaran visual dari pengguna dan administrator dalam platform e-voting.

3) Implementasi

Tahap ini merupakan tahap penting dalam pembuatan sistem e-voting berbasis *blockchain*. Implementasi mencakup pengkodean dan pengujian untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi yang telah dirancang pada tahap workshop design. Dengan beberapa tahap yang ada. Pada tahap awal implementasi, fokus utama adalah pada pengembangan sistem e-voting. Pengembangan ini mencakup beberapa aspek penting yaitu, pembuatan *smart contract*, pengkodean sistem, integrasi *blockchain*.

III. HASIL DAN PEMBAHASAN

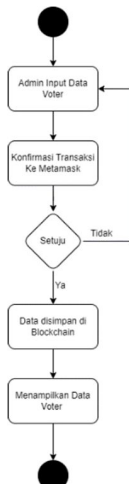
BAB III ini penulis akan memaparkan hasil dari penelitian yang telah dilakukan mengenai penerapan teknologi *blockchain* untuk meningkatkan transparansi dan keamanan dalam sistem e-voting.

A. Proses Bisnis Aplikasi

Proses bisnis dalam aplikasi ini mencakup fungsi utama dalam aplikasi yang akan dibuat yaitu mulai dari registrasi voter, dan bagaimana proses pemungutan suara. Setiap tahap akan didesain untuk menjaga integritas data dan memberikan pengalaman pengguna yang optimal.

1) Registrasi Voter

Voter mengisi formulir registrasi dengan data pribadi yang diperlukan (diluar sistem aplikasi), Data voter dikirim ke admin untuk verifikasi (diluar sistem aplikasi), Admin melakukan transaksi data ke *blockchain* untuk memverifikasi data voter, Jika verifikasi berhasil, admin memberikan hak suara kepada pemilih, Voter-voter yang terverifikasi akan ditampilkan ke aplikasi.



Gambar 2 Activity Diagram Admin Input Data Voter



Gambar 3 Activity Diagram Proses Pemungutan Suara

2) Pemungutan Suara

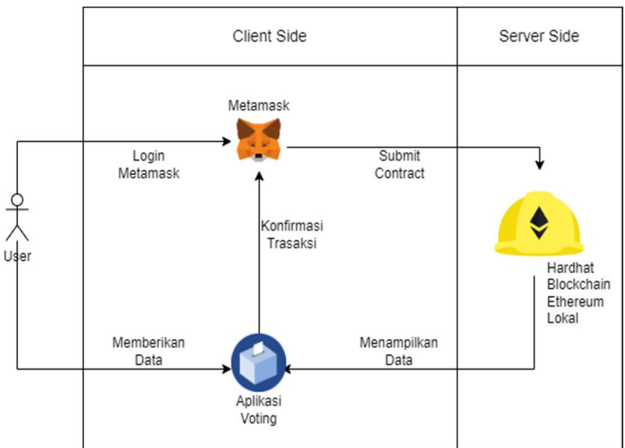
Sistem melakukan verifikasi hak akses pemilih apakah sudah teregistrasi atau tidak, Jika sudah teregistrasi pemilih memiliki hak untuk memilih candidate yang diinginkan, Suara pemilih disimpan di *blockchain*, Sistem akan mengembalikan data yang sudah tersimpan di *blockchain* ke aplikasi.

B. Desain Arsitektur

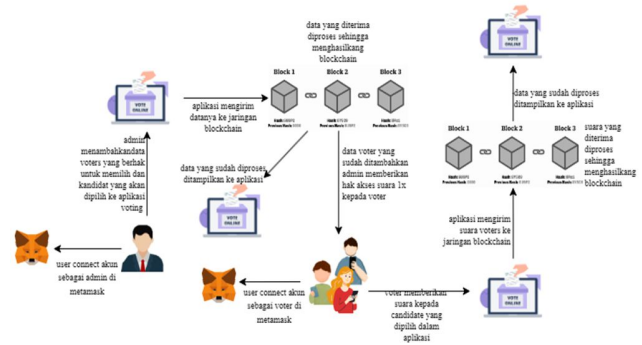
Dalam penelitian ini, penulis merancang beberapa hal dalam menunjang aplikasi, sehingga membantu juga dalam pembuatan aplikasi, perancangan yang ada seperti desain *activity diagram*, desain arsitektur sistem juga penulis menambahkan *rich picture* dalam mempermudah pemahaman memberikan deskripsi luas tentang suatu situasi[16] untuk memahami bagaimana sistem *e-voting* berbasis *blockchain* bekerja.

Activity diagram pada gambar 2 menunjukan proses penginputan data disitu terlihat admin menginput data voter dan untuk mengirim data ke jaringan *blockchain* admin harus mengkonfirmasi dulu transaksi dalam pengiriman data voter ke

Gideon L. Saroinsong – Penerapan Teknologi Blockchain Pada Sistem E-Voting



Gambar 4 Desain Arsitektur Sistem



Gambar 5 Rich Picture

blockchain, jika admin tidak setuju akan kembali ke tampilan penginputan data, jika admin setuju data candidate berhasil disimpan ke jaringan *blockchain* lewat akun *metamask*, setelah *blockchain* menerima data, menampilkan data ke aplikasi voting.

Activity diagram pada gambar 3 menunjukkan proses pemungutan suara dalam gambar yang ditampilkan voter masuk ke halaman home selanjutnya voter mengkoneksikan akun *metamask* dengan akun yang terdaftar setelah itu dalam aplikasi home menampilkan data candidate selanjutnya voter bisa memilih siapa candidate yang ingin dipilih, setelah memilih candidate voter melakukan konfirmasi ke *metamask* jika setuju dan jika akun yang terkoneksi ke *metamask* adalah akun yang sudah terdaftar maka suara candidate yang dipilih akan tersimpan di *blockchain*, selanjutnya *blockchain* akan menampilkan hasil voting terbaru yang sudah terupdate.

Pada gambar 4 menjelaskan bagaimana desain arsitektur sistem. Pertama pengguna memulai dengan membuka aplikasi voting di perangkat mereka, untuk melakukan voting, pengguna harus terlebih dahulu login ke dompet digital mereka dalam hal ini penulis menggunakan *Metamask*. Setelah login ke *MetaMask*, pengguna dapat mengakses antarmuka aplikasi *e-voting*. Di sini, pengguna memasukkan data yang diperlukan sesuai dengan fitur yang diberikan pada *e-voting*, seperti kalau admin mereka bisa menambahkan baik peserta voter maupun menambahkan candidate, begitu dengan voter mereka bisa memilih candidate yang diinginkan dengan catatan mereka bisa memilih ketika akun mereka sudah terkonfirmasi oleh admin. Data-data tersebut kemudian disiapkan untuk dikirimkan

TABEL I
PENGUJIAN FUNGSI VOTERRIGHT

PENGIRIM	FUNGSI	INPUT	HASIL
Akun 3 (Bukan Admin)	<i>voterRight</i>	(address akun 4, data voter)	Gagal ("Anda tidak memiliki otoritas untuk menambahkan Voter")
Akun 1 (Admin)	<i>voterRight</i>	(address akun 4, data voter)	Berhasil

TABEL II
PENGUJIAN FUNGSI SETCANDIDATE

PENGIRIM	FUNGSI	INPUT	HASIL
Akun 4 (Bukan Admin)	<i>setCandidate</i>	(address akun 7, data voter)	Gagal ("Anda tidak memiliki otoritas untuk menambahkan Voter")
Akun 1 (Admin)	<i>setCandidate</i>	(address akun 7, data voter)	Berhasil

sebagai transaksi di *blockchain*. Setelah user memasukkan data yang dilakukan, aplikasi e-voting meminta konfirmasi dari pengguna untuk menyetujui transaksi di *Metamask*. user akan melihat pop-up dari *Metamask* yang meminta mereka untuk mengonfirmasi detail transaksi. Ini merupakan langkah keamanan penting untuk memastikan bahwa transaksi yang dikirim ke *blockchain* memang benar diotorisasi oleh user. Setelah transaksi dikonfirmasi oleh user di *Metamask*, transaksi tersebut dikirim ke jaringan *blockchain* lokal yang dijalankan *Hardhat*. Setelah transaksi diproses jaringan *blockchain* (*Hardhat*), hasilnya dikembalikan ke aplikasi e-voting. Aplikasi kemudian menampilkan data ini kepada seluruh pengguna yang masuk ke aplikasi, seperti konfirmasi bahwa suara mereka telah diterima dan dicatat dengan benar di *blockchain*.

C. Pengujian Fungsi Smart Contract

Pada tabel 1 akun 3 penulis melakukan pengujian penambahan voter pada *smart contract* dan saat memasukkan data pada fungsi *voterRight* data tidak berhasil dilakukana atau transaksi gagal dengan mengembalikan data yaitu "Anda tidak memiliki otoritas untuk menambahkan Vote", dikarenakan account *0xAb8...* bukan admin, sedangkan pada tabel 1 akun 1 penulis melakukan pengujian penambahan voter pada *smart contract* dengan data yang sama yang diberikan pada pengujian sebelumnya dan saat melakukan transaksi data berhasil dengan input dan output yang diberikan bernilai sama, dikarenakan account *0x5B3...* yang melakukan transaksi adalah address yang terdaftar sebagai admin.

Dengan kesimpulan hasil yang di dapatkan pada tabel 1 yaitu hanya akun dengan hak admin (Akun 1) yang di dapatkan pada pengujian sesuai harapan

Pada tabel 2 penulis melakukan pengujian penambahan candidate pada *smart contract* dan saat memasukkan data pada

TABEL III
PENGUJIAN FUNGSI VIEW

FUNGSI view	HASIL
<i>getCandidate()</i>	Data berisi <i>address-address</i> kedua <i>candidate</i>
<i>getCandidateLength()</i>	Angka 2 (jumlah <i>candidate</i>)
<i>getCandidateData(address_candidate_1)</i>	Berisi data <i>candidate</i> 1 (usia, nama, <i>ID</i> , gambar, jumlah suara, <i>IPFS</i> , <i>address</i>)
<i>getCandidateData(address_candidate_2)</i>	Berisi data <i>candidate</i> 2 (usia, nama, <i>ID</i> , gambar, jumlah suara, <i>IPFS</i> , <i>address</i>)
<i>getVoterList()</i>	Data berisi <i>address</i> pemilih
<i>getVoterLength()</i>	Angka 1 (jumlah pemilih)
<i>getVoterData(address_pemilih)</i>	berisi data pemilih (<i>ID</i> , nama, gambar, <i>address</i> , <i>IPFS</i> , status <i>allowed</i> , status <i>voted</i>)

fungsi *setCandidate* data tidak berhasil dilakukana atau transaksi gagal dengan mengembalikan data yaitu "Anda tidak memiliki otoritas untuk menambahkan Candidate", dikarenakan account *0x4B2...* bukan admin, sedangkan pada tabel 2 penulis melakukan pengujian penambahan voter pada *smart contract* dengan data yang sama yang diberikan pada pengujian sebelumnya dan saat melakukan transaksi data berhasil dengan input dan output yang diberikan bernilai sama, dikarenakan account *0x5B3...* yang melakukan transaksi adalah address yang terdaftar sebagai admin.

Dengan kesimpulan hasil yang di dapatkan pada tabel 1 dan tabel 2 data yang menampilkan yaitu hanya akun dengan hak admin (Akun 1) yang di dapatkan pada pengujian sesuai harapan.

Pada tabel 3 penulis melakukan pengujian fungsi *getCandidate()* dan *getCandidateLength()* dengan output pengujian berhasil, dimana untuk fungsi *getCandidate()* yaitu mengambil *address* data yang sudah disimpan ke *blockchain* dari fungsi *setCandidate*. Sedangkan untuk fungsi *getCandidateLength()* menampilkan berapa jumlah *candidate* yang terdaftar yang sudah disimpan ke *blockchain* dari fungsi *setCandidate*. Penulis juga melakukan pengujian fungsi *getCandidateData* (*address candidate* 1 dan 2) dan setelah dipanggil data berhasil berisi data *candidate* 1 dan 2 antara lain usia, nama, *ID*, gambar, jumlah suara, *IPFS*, *address* walaupun untuk gambar dan *ipfs* menampilkan data string dikarenakan *remix* tidak mendukung output tampilan gambar.

Penulis juga melakukan pengujian fungsi *getVoterList()* dan *getVotersLength()* dengan output pengujian berhasil, dimana untuk fungsi *getVoterList()* yaitu mengambil *address* data yang sudah disimpan ke *blockchain* dari fungsi *voterRight*.

```
PS C:\Users\HEWLETT\PACKARD\OneDrive\Dokumen\blockchain\fixoo\e-voting\fixoo> npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
=====

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238f944abc478cbed5fcae784d7bf4f2ff80

Account #1: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a004a966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC (10000 ETH)
Private Key: 0x5de4111fa1a4b94908f83103eb1f1706367c2e68ca8870fc3fb9a804cdab365a

Account #3: 0x90f79bfe6B2c4f80365E785982E1f01E93b906 (10000 ETH)
Private Key: 0x7c852118294e51653712a81e58004919141751be58f605c371e15141b007a6

Account #4: 0x15d34AAf542670D707c36783AAf7A00a2C6A65 (10000 ETH)
Private Key: 0x47e179ec197488993b187f80a00e0bda91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507D1a55bc2695C58ba16F37d81980A4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca88899a2d4f0ec3153be03185e2d3348e72892edfba

Account #17: 0xbDA5747bD65F08deb54cb465eB87D40e51B197E (10000 ETH)
Private Key: 0x689af8efa8c651a91ad287602527f3af2fe9f6501a7ac4b061667b5a93e037fd

Account #18: 0xd02FD4581271e230360230F9337D5c04308f44C0 (10000 ETH)
Private Key: 0xde9be858da4a475276426320d5e9262ecf3ba460bfac56360bfa6c428b4ee0

Account #19: 0x8626f6940E2eb28930Fb4CeF49B2d1F2C9C1199 (10000 ETH)
Private Key: 0xdf57889febbacf7ba0bc227dafbffa9fc08a93fcd68e142411a14efcf23656e

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

eth_blockNumber
eth_getBlockByNumber
net version (2)
```

Gambar 6. Tampilan Saat Memasuki Node Blockchain Hardhat

```
PS C:\Users\HEWLETT\PACKARD\OneDrive\Dokumen\blockchain\fixoo\e-voting\fixoo> npx hardhat run scripts/deploy.js
--network localhost
CONTRACT_ADDRESS: 0x5fbd2315678afecb367f032d93f642f64180aa3
PS C:\Users\HEWLETT\PACKARD\OneDrive\Dokumen\blockchain\fixoo\e-voting\fixoo>
```

Gambar 7. Tampilan Saat Smart Contract di Deploy ke Blockchain Hardhat

Sedangkan untuk fungsi `getVotersLength()` menampilkan berapa jumlah voter yang terdaftar yang sudah disimpan ke *blockchain* dari fungsi `voterRight`. Penulis juga melakukan pengujian fungsi `getVoterData` dan `get VotedVotersList` dan setelah dipanggil fungsi `getVoterData` data berhasil berisi data voter antara lain ID, nama, gambar, address, IPFS, status allowed, status voted, untuk fungsi `getVotedVotersList` menampilkan output yaitu addres voter yang sudah melakukan vote, dengan kesimpulan pengujian pada tabel 3 pada pengujian fungsi view sesuai harapan.

D. Implementasi Sistem

Pada gambar 6 ini menampilkan gambar saat memasuki node *blockchain hardhat* dengan perintah `npx hardhat node`, Perintah ini memulai node *blockchain* Ethereum lokal yang berjalan pada jaringan pribadi. Node ini meniru perilaku jaringan Ethereum sehingga bisa melakukan pengembangan dan pengujian., dan juga memberikan 20 account yang sudah diisi dengan Ether (ETH) palsu untuk

Pada bagian ini menampilkan gambar saat memasuki node *blockchain hardhat* dengan perintah `npx hardhat node`, Perintah ini memulai node *blockchain* Ethereum lokal yang berjalan pada jaringan pribadi. Node ini meniru perilaku jaringan Ethereum sehingga bisa melakukan pengembangan dan pengujian., dan juga memberikan 20 account yang sudah diisi dengan Ether (ETH) palsu untuk digunakan dalam pengembangan dan pengujian.

Pada gambar 7 ini menampilkan gambar saat *smart contract* berhasil di deploy dengan perintah `npx hardhat run scripts/deploy.js --network localhost`, pada perintah tersebut

```
PS C:\Users\HEWLETT\PACKARD\OneDrive\Dokumen\blockchain\fixoo\e-voting\fixoo> npm run dev
> votingproject@0.1.0 dev
> next dev

ready - started server on 0.0.0.0:3000, url: http://localhost:3000
Browserslist: caniuse-lite is outdated. Please run:
  npx browserslist@latest --update-db
  Why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
event - compiled client and server successfully in 8.2s (448 modules)
wait - compiling / (client and server)...
event - compiled client and server successfully in 700 ms (476 modules)
wait - compiling /candidate-registration...
event - compiled client and server successfully in 713 ms (503 modules)
```

Gambar 8. Tampilan Saat Sistem berjalan di local host

```
eth_call
Contract call: Create#getVoterList
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b92266
To: 0x5fbd2315678afecb367f032d93f642f64180aa3

eth_call
Contract call: Create#getVoterData
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b92266
To: 0x5fbd2315678afecb367f032d93f642f64180aa3

eth_call
Contract call: Create#getVoterData
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b92266
To: 0x5fbd2315678afecb367f032d93f642f64180aa3

eth_call
Contract call: Create#getVoterData
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b92266
To: 0x5fbd2315678afecb367f032d93f642f64180aa3

eth_call
Contract call: Create#getVoterLength
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b92266
To: 0x5fbd2315678afecb367f032d93f642f64180aa3
```

Gambar 9. Tampilan Saat Smart Contract di panggil di aplikasi

```
4081871793aaeaa00786a811aba819698273c132cae
Block Number: 5
Block Hash: 0xadcb60c3605c57fbabd131fc93a262024c261ced58fbd14035dcb1918397b247
Block Number: 6
Block Hash: 0x899bfbf81e0bbe91ef2d3a7e041d861d15191d6a8ee83dde748ecd970a9cc8c9
Block Number: 7
Block Hash: 0x2b546b37a87c9a6df1fbd5ca79e1aef5d00f53a02ae27b612856c073647d9adc
Block Number: 8
Block Hash: 0x8aefad4eb183811842ca3b59f68143cae1e41fd1f7c28fd8d06096200c8049
```

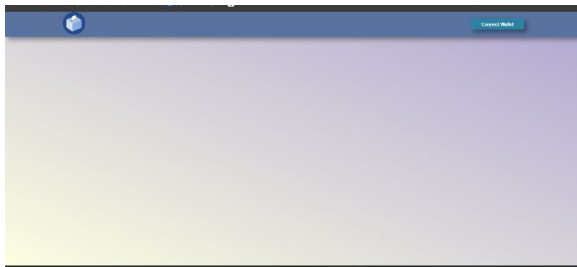
Gambar 10. Tampilan Jumlah Block Number dan Hash

hardhat dengan secara otomatis mengcompile *smart contract* yang ada dalam coding tersebut dan terhubung ke node *blockchain* lokal yang sedang berjalan di *localhost*, dan outputnya menampilkan *smart contract* yang telah di-deploy yaitu alamat contract.

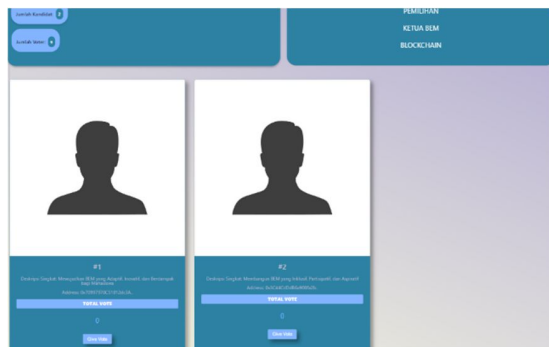
Pada bagian ini menampilkan bahwa sistem berhasil berjalan di jaringan *blockchain* di *localhost* dengan perintah `npm run dev`, perintah tersebut digunakan untuk menjalankan aplikasi dalam mode pengembangan, mode ini digunakan untuk mengembangkan dan menguji aplikasi secara lokal sebelum di-deploy ke jaringan utama.

Pada gambar 9 bagian ini menunjukkan log atau output dari panggilan aplikasi ke *smart contract*, dengan salah satu contoh yaitu `getVoterList` Fungsi untuk mendapatkan daftar pemilih. `getVoterData` Fungsi untuk mendapatkan data pemilih. `getVoterLength` Fungsi untuk mendapatkan jumlah daftar pemilih.

Pada gambar 10 bagian ini menunjukkan jumlah block number dan *Hash* yang sudah ada pada *blockchain* lokal. Block number adalah angka yang menunjukkan data blok dari sebuah rantai blok (*blockchain*). *Hash* dari blok dihasilkan melalui algoritma kriptografi. *Hash* ini berfungsi sebagai identifikasi unik untuk setiap blok. Contohnya, blok nomor 5 memiliki



Gambar 11 Tampilan Awal Aplikasi



Gambar 12 Tampilan Menu Home



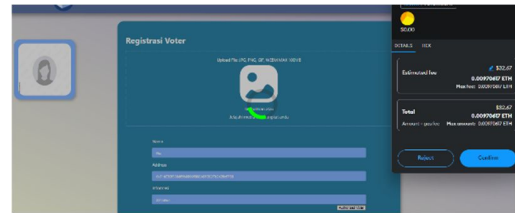
Gambar 13 Tampilan Menu Tambahkan Candidate

Hash *0xad6b0c3605c57fbabd131fc93...397b247*. Setiap blok mengandung informasi transaksi dan data lainnya yang telah dicatat pada *blockchain*. Hash setiap blok dibuat berdasarkan konten blok itu sendiri dan Hash dari blok sebelumnya, menciptakan rantai yang aman dan terhubung secara *kriptografis*. Hal ini memastikan bahwa setiap perubahan pada satu blok akan mempengaruhi Hash dari semua blok berikutnya, sehingga membuat *blockchain* sulit untuk dimanipulasi.

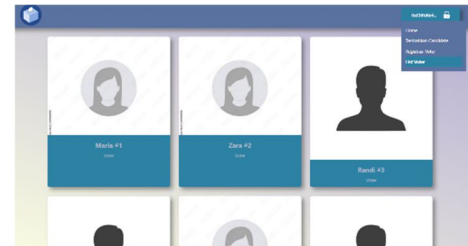
Pada gambar 11 merupakan tampilan awal sistem menunjukan tampilan yang bersih yang hanya ada logo dan button Connect Wallet. Pada tampilan ini fokus untuk menghubungkan dompet user lewat button connect wallet *metamask* dengan aplikasi sistem yang dibuat.

Gambar 12 merupakan bagian ini menampilkan kartu informasi yang memuat jumlah candidate dan jumlah voter yang sudah terdaftar dalam sistem. Setiap candidate ditampilkan dalam bentuk kartu yang berisi foto candidate, nama lengkap, dan jumlah suara yang telah diterima oleh masing-masing candidate. Selain itu, terdapat tombol "Give Vote" di bawah setiap kartu candidate yang memungkinkan voter untuk memberikan suara mereka kepada candidate yang dipilih.

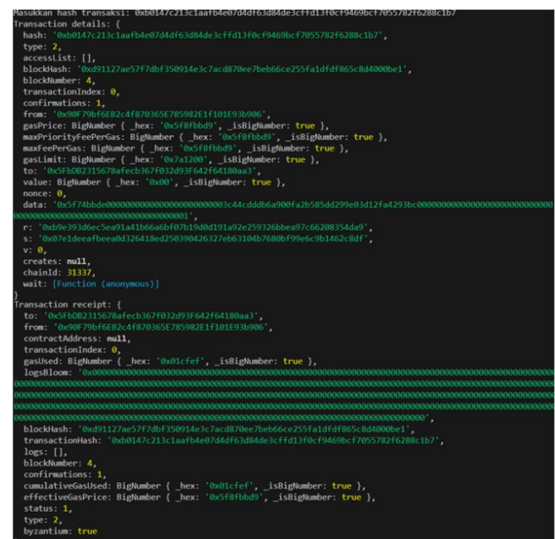
Gambar 13, Pada bagian ini tampilan antarmuka user yang digunakan untuk menambah voter dalam aplikasi. Bagian utama halaman ini terdiri dari formulir yang harus diisi admin



Gambar 14 Tampilan Menu Tambahkan Voter



Gambar 15 Tampilan Menu List Voter



Gambar 16 Tampilan Pengecekan Hash

Dan *metamask* untuk mengkonfirmasi transaksi data yang sudah di input dan akan di masukkan ke jaringan *blockchain*.

Gambar 14, terdiri dari sebuah formulir yang dirancang untuk menambahkan candidate dengan data-data yang diperlukan informasi mengenai candidate baru.

Gambar 15, bagian ini merupakan tampilan antarmuka user untuk melihat voter-voter yang sudah teregistrasi admin.

Pada bagian ini menampilkan hasil dari pengecekan transaksi Hash saat voter sudah memberikan suara kepada kandidat yang mereka pilih. Transaksi ini memberikan informasi bagaimana detail setiap transaksi yang terjadi di dalam sistem e-voting berbasis *blockchain* yang dapat dipantau dan diverifikasi secara transparan dan aman. Transaksi yang dihasilkan memiliki Hash *0xb0147c213c1aafb4e07d...5782f6288c1b7*, yang berfungsi sebagai identitas unik untuk transaksi tersebut. Informasi yang terdapat di dalam transaksi ini dalam gambar menampilkan address pengirim yaitu *0x90F79bf6EB2c4...101E93b906*, dan address penerima yaitu *0x5FbDB2315678afec...2f64180aa3*. Selain itu, memberikan juga detail seperti *blockHash* yang menunjukkan Hash dari blok yang berisi transaksi ini (*0xd91127ae57f7dbf...e7beb66ce255faldfdf865c8d4000be1*),

blockNumber yang menunjukkan nomor blok (4), dan gasUsed yang menunjukkan jumlah gas yang digunakan (0x01cfef). Dan informasi ini juga dapat diverifikasi oleh siapa saja, sehingga menjamin keamanan dan transparansi dari sistem e-voting ini.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian dan implementasi teknologi *blockchain* dalam sistem e-voting, kesimpulan yang diperoleh adalah sebagai berikut:

1. Setiap fungsi pada *smart contract* dilakukan pengujian hasil berfungsi dan berjalan dengan baik.
2. Setiap voter hanya dapat memberikan satu suara, yang mencegah kecurangan seperti penggandaan suara atau pemalsuan identitas.
3. Aplikasi yang dibuat dengan berbasis *blockchain* voter bisa Cross check suara mereka terkirim dan tersimpan di *blockchain* dan masih tetap sama atau tidak lewat *Hash* yang didapatkan lewat *metamask*.

B. Saran

1. Pengembangan fitur keamanan tambahan seperti verifikasi biometrik, seperti sidik jari atau pengenalan wajah.
2. Dilakukan pengujian lebih lanjut untuk memastikan sistem dapat menangani jumlah pemilih yang lebih besar
3. Perlunya edukasi pihak yang akan menggunakan aplikasi ini
4. Kerjasama dengan pemerintah atau institusi untuk mengintegrasikan sistem ini ke dalam proses pemilihan resmi.
5. Sistem e-voting harus terus dipantau dan diperbarui untuk menghadapi ancaman keamanan yang terus berkembang.

V. KUTIPAN

- [1] J. A. Halderman And V. Teague, "The New South Wales Ivote System: Security Failures And Verification Flaws In A Live Online Election," Apr. 2015, [Online]. Available: <http://arxiv.org/abs/1504.05646>
- [2] V. Primadyanti, J. Waleng, G. Putu, K. Juliharta,) Ketut, And Q. Fredlina, "Implementasi *E-Voting* Pemilu Raya Mahasiswa Universitas Primakara Berbasis Web (Studi Kasus Pada Universitas Primakara)," Pp. 464–473.
- [3] V. Taniady, B. Aditya Prawira Arafat, And H. Sutra Disemadi, "Sistem E-Voting Dalam Pemilihan Kepala Daerah 2020 Saat Pandemi Covid-19: Perbandingan Indonesia, Australia Dan Brazil," Vol. 19, No. 2, Pp. 1055–1064, 2020.
- [4] M. A. Lubis, M. Y. A. Gea, And N. Muniifah, "Penerapan Asas Pemilu Terhadap Electronic Voting (*E-Voting*) Pada Pemilu Tahun 2024," *Jurnal Ilmiah Penegakan Hukum*, Vol. 9, No. 1, Pp. 44–56, Jun. 2022, Doi: 10.31289/jiph.V9i1.6491.
- [5] S. D. K. Hu, H. N. Palit, And A. Handojo, "Implementasi *Blockchain*: Studi Kasus E-Voting," *Jurnal Infra*, Vol. 7, No. 1, Pp. 183–189, 2019.
- [6] K. Mehboob Khan, J. Arshad, And M. M. Khan, "Secure Digital Voting System Based On *Blockchain* Technology," 2023.
- [7] T. Enades Hari Setia And S. Ajib, "Smart Contract *Blockchain* Pada *E-Voting*," *Informatika Upgris*, Vol. 5, No. 2, Pp. 188–191, 2019.
- [8] A. F. Prasetyo And R. Munir, "Aplikasi Voting Online Dengan Menggunakan Teknologi *Blockchain*," No. M, Pp. 1–5, 2020.
- [9] Wijaya Junior Hendri, A. Zulfikar, And I. Permatasari, "Implementasi Sistem *E-Voting* Untuk Meningkatkan Kualitas Demokrasi Di Indonesia," *3universitas Muhammadiyah Yogyakarta*, Pp. 51–57, 2019.
- [10] T. Wira And E. Suryawijaya, "Memperkuat Keamanan Data Melalui Teknologi *Blockchain*: Mengeksplorasi Implementasi Sukses Dalam Transformasi Digital Di Indonesia Strengthening Data Security

- Through *Blockchain* Technology: Exploring Successful Implementations In Digital Transformation In Indonesia," Vol. 2, No. 1, Pp. 55–67, 2023, Doi: 10.21787/jskp.2.2023.55-67.
- [11] A. M. Maburoh, F. Dewanta, And A. A. Wardana, "Implementasi *Ethereum Blockchain* Dan *Smart Contract* Pada Jaringan Smart Energy Meter," 2021.
- [12] A. Kusyanti, P. Hari Trisnawan, And P. Korespondensi, "Implementasi Teknologi *Blockchain* Dengan Sistem *Smart Contract* Pada Klaim Asuransi," Vol. 10, No. 7, Pp. 1563–1570, 2023, Doi: 10.25126/Jtiik.2023108016.
- [13] Kennedy, "Implementasi Teknologi *Blockchain* Untuk Pengiriman Jawaban Cbt Menggunakan Rsa-Sha256," *Universitas Hasanuddin*, 2021.
- [14] H. S. Kartiko, "Implementasi Ipfs Untuk Mengurangi Gas Fee *Smart Contract* *Ethereum* Pada Aplikasi Penggalangan Dana," *Jepin (Jurnal Edukasi Dan Penelitian Informatika)*, Vol. 9, No. 2, Pp. 195–203, 2023.
- [15] D. Hariyanto, R. Sastra, F. E. Putri, S. Informasi, K. Kota Bogor, And T. Komputer, "Implementasi Metode Rapid Application Development Pada Sistem Informasi Perpustakaan," 2021.
- [16] R. Taufik, P. Citra Dewi, K. Widina, And A. Anwar, "Analisis Banjir Di Kota Bandung Dengan Pemodelan Sistem *Rich Picture Diagram*," [Online]. Available: <https://bandungkota.bps.go.id/>



Gideon Lukas Saroinsong. Lahir di Sapa, 17 Mei 2003 dari pasangan Alm. Royke Saroinsong dan Hesty Ngongoloy. Penulis merupakan anak ketiga dari tiga bersaudara. Penulis tinggal di Sapa Barat, Kec. Tenga, Kab. Minahasa Selatan, Prov. Sulawesi Utara. Penulis menempuh pendidikan dari taman kanak-kanak di TK GMIM Tiberias Sapa pada tahun 2007-2008. Pendidikan SD di SD Inpres Sapa pada tahun 2008-2014. Pendidikan SMP di SMP Negeri 3 Tenga pada tahun 2014-2017. Pendidikan SMA di SMA Negeri 1 Tenga pada tahun 2017-2020. Setelah lulus dari SMA pada tahun 2020 penulis melanjutkan pendidikan Strata satu (S1) di Universitas Sam Ratulangi Manado mengambil Program Studi Teknik Informatika Jurusan Teknik Elektro Fakultas Teknik. Selama masa perkuliahan, penulis tergabung dalam organisasi kemahasiswaan, yaitu Himpunan Mahasiswa Elektro dan penulis juga tergabung dalam Unit Pelayanan Kristen Fakultas Teknik.