Analisis Keamanan Sistem *E-Voting* Berbasis *Blockchain* dengan Menggunakan Ganache dan Echidna

Peter Reynard Susanto[1], Muhammad Rizky Pribadi[2]
[1,2]Program Studi Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang,
Jl. Rajawali No. 14, Palembang, Indonesia.
peterreynardsusanto_2226250020@mhs.mdp.ac.id[1] , rizky@mdp.ac.id[2]

Abstrak. *E-voting merupakan sebuah proses pemungutan suara yang dilakukan dengan bantuan media teknologi informasi yang bertujuan untuk mempercepat serta mempermudah proses pemungutan dan perhitungan suara pada kegiatan pemilihan umum. Namun, penerapan e-voting masih menghadapi berbagai tantangan, terutama terkait aspek keamanan dan integritas data pemilihan suara karena keamanan merupakan aspek yang krusial dalam proses demokratis. Sistem e-voting berbasis blockchain menawarkan transparansi dan integritas data, tetapi masih menghadapi tantangan keamanan khususnya pada smart contract yang merupakan inti dari proses pemungutan suara. Penelitian ini bertujuan untuk menguji tingkat keamanan smart contract pada e-voting berbasis blockchain Ethereum dengan metode pengujian berbasis properti menggunakan fuzzer Echidna. Pengujian dilakukan terhadap lima properti keamanan, yaitu tidak bisa double vote, total suara tidak pernah berkurang, jumlah kandidat selalu tetap, jumlah suara dan pemilih sama serta pemilih selalu memberikan satu suara. Hasil pengujian menunjukkan bahwa seluruh properti dinyatakan passing karena tidak ditemukan pelanggaran logika meskipun Echidna telah melakukan ribuan transaksi acak untuk menemukan bug.*

Kata Kunci: *Blockchain, Echidna, Ethereum, E-Voting, Smart Contract.*

Abstract. *E-voting is a voting process conducted using information technology, aimed at accelerating and simplifying the voting and vote counting process in general elections. However, the implementation of e-voting still faces various challenges, particularly related to the security and integrity of voting data, as security is crucial in the democratic process. Blockchain-based e-voting systems offer transparency and data integrity, but still face security challenges, particularly regarding smart contracts, which are at the heart of the voting process. This study aims to test the security level of smart contracts in Ethereum blockchain-based e-voting using a property-based testing method using the Echidna fuzzer. Testing was conducted on five security properties: no double voting, the total number of votes never decreases, the number of candidates is always constant, the number of votes and voters is the same, and voters always cast one vote. The test results showed that all properties were declared passing because no logic violations were found even though Echidna had performed thousands of random transactions to find bugs.*

Keywords: *Blockchain, Echidna, Ethereum, E-Voting, Smart Contract.*

PENDAHULUAN

Pemungutan suara adalah hal yang penting dalam negara demokrasi. Sebelum berkembangnya teknologi informasi, pemungutan suara masih dilakukan secara manual. Pemungutan suara secara manual ini dilakukan dengan cara pemilih yang mempunyai hak pilih datang langsung ke tempat pemilihan dan melakukan pencoblosan pada kertas suara kemudian memasukannya ke kotak suara. Setelah pemilihan selesai, perhitungan suara dilakukan secara manual sehingga berjalan lambat. Pemilihan dengan cara manual seringkali terjadi kesalahan yang disebabkan oleh banyak faktor, seperti banyaknya daftar pemilih ganda, ada pemilih yang memilih lebih dari satu pasangan calon yang mengakibatkan surat suara tidak sah, dan juga banyak kertas suara yang cacat atau rusak [1]. Hal ini tentu saja kurang efektif. Pemungutan suara seperti ini selain membutuhkan biaya yang tidak sedikit juga memerlukan waktu yang cukup banyak mulai dari pemilihan sampai penghitungan suara.

Tetapi dengan berkembangnya teknologi informasi, pemungutan suara atau voting ditingkatkan dengan menggunakan *electronic voting*. *Electronic voting* atau sering disebut *e-voting*

merupakan sebuah proses pemungutan suara yang dilakukan dengan bantuan media teknologi informasi yang memiliki tujuan untuk mempercepat serta mempermudah proses pemungutan dan perhitungan suara pada kegiatan pemilihan umum, serta dapat menggantikan kertas suara [2]. Dengan *e-voting*, pemilih tidak perlu datang ke lokasi pemilihan, melainkan hanya cukup melakukan pemilihan menggunakan *smartphone* atau komputer masing-masing. Dengan *e-voting* kita dapat langsung melihat hasil pemilihan yang sedang berjalan dan dapat memantau calon mana yang lebih unggul. Hal ini jauh lebih efektif dari sistem pemilihan manual yang dilakukan sebelumnya, baik dari segi biaya maupun waktu.

Namun permasalahan utama dalam sistem *e-voting* adalah data disimpan di dalam *database* terpusat yang dikelola oleh salah satu pihak, sehingga membuat sistem rentan terhadap serangan siber dan manipulasi data oleh pihak yang tidak berwenang [3]. Oleh karena itu diperlukan suatu teknologi yang dapat menjamin keamanan *e-voting*.

Teknologi yang ada terus mengalami perkembangan, sehingga saat ini dikenal suatu teknologi yang disebut *blockchain*. *Blockchain* pada dasarnya adalah buku besar yang terdistribusi dari semua transaksi yang dilakukan secara langsung antara konsumen dan penyedia pada sistem [4][5]. Teknologi *blockchain* dapat digunakan dalam berbagai bidang seperti bidang keuangan, kesehatan, dan pendidikan, termasuk *e-voting.* Implementasi *e-voting* berbasis *blockchain* di dunia nyata telah dieksplorasi di berbagai negara, seperti Estonia dan Swiss, yang menunjukkan potensi *blockchain* untuk meningkatkan keamanan dan transparansi pemilu [6]. Dengan menggunakan buku besar terdesentralisasi yang terdistribusi di seluruh jaringan, *blockchain* memungkinkan pencatatan suara yang aman, tak terubah, dan terverifikasi. Setiap transaksi suara dicatat dalam blok-blok yang saling terhubung, dan data tidak dapat dimanipulasi tanpa persetujuan jaringan secara luas. Ini menghasilkan tingkat keamanan yang tinggi dan memungkinkan pemilih dan pihak berkepentingan untuk memverifikasi hasil pemilihan dengan lebih mudah [7].

Ethereum sebagai salah satu jaringan *blockchain* memiliki jaringan komputer yang sudah mapan. Ethereum menggunakan algotitma PoS (*Proof of Stake*) yaitu algoritma konsensus yang memilih validator berdasarkan jumlah koin yang dimiliki. Semakin banyak koin yang dimiliki, semakin besar peluang seseorang untuk dipilih sebagai validator yang memvalidasi transaksi dan membuat blok baru [8]. Platform Ethereum juga memiliki bahasa pemrograman *Ethereum Virtual Machine* dan Solidity. Solidity dapat digunakan untuk membuat aplikasi terdesentralisasi atau kontrak pintar yang kemudian dikompilasi oleh Mesin Virtual Ethereum dan dijalankan di *blockchain* [9][10][11].

Smart contract yang ditulis dengan bahasa pemrograman Solidty harus diperiksa dengan benar agar tidak ada *bug*, karena kesalahan kode pada *smart contract* dapat menyebabkan program tidak aman dan rentan dari serangan pihak yang tidak bertanggung jawab untuk merusak sistem. *Smart contract* tidak dapat diubah apabila sudah dijalankan di *blockchain* karena sifatnya yang kekal *(Immutability).* Oleh karena itu diperlukan suatu tool yang dapat melakukan pengujian terhadap *smart contract* yang telah dibuat agar terhindar dari kesalahan kode yang akhirnya dapat menyebabkan kerentanan sistem. Apabila sampai terjadi bug, kerentanan pada sistem dapat dieksploitasi untuk melakukan *fraud* yang memengaruhi integritas data *voting*.

Berdasarkan latar belakang di atas, penulis bermaksud menganalisa sistem keamanan *e-voting* berbasis *blockchain*. Penelitian sebelumnya mengkaji penerapan teknologi *blockchain* dalam sistem *e-voting*, mengidentifikasi tantangan yang ada seperti transparansi yang kurang, kertas suara yang rusak, kekurangan saksi, hingga permasalahan dalam perhitungan suara, dan menawarkan solusi yang dapat mengatasi masalah tersebut [12].

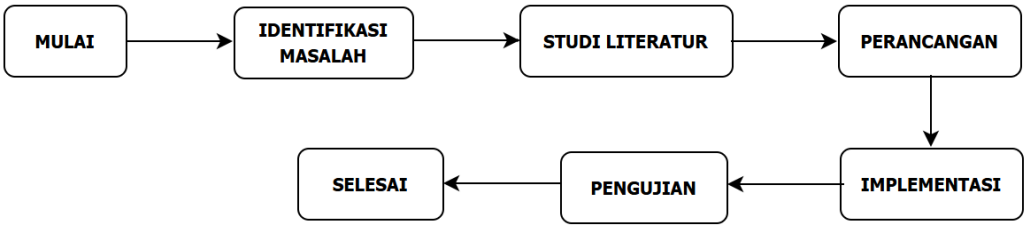
Penelitian terdahulu mengenai *e-voting* berbasis *blockchain* dengan menggunakan jaringan lokal hardhat [13] lebih fokus menjelaskan bagaimana cara melakukan *e-voting* dan arti dari fungsi kode yang ada di *smart contract*. Penelitian Aplikasi *E-voting* Berbasis *Blockchain* dengan Metode *Smart Contract* [2], menggunakan metode Binance *smart chain* dan lebih berfokus pada cara melakukan voting. Implementasi *Smart Contract Blockchain* Ethereum Pada Aplikasi E-voting [5]

menggunakan jaringan Ethereum Sepolia untuk mengimplementasikan *smart contract blockchain* Ethereum pada aplikasi *e-voting* dengan tujuan meningkatkan keamanan, integritas, dan transparansi proses pemilihan, lebih berfokus pada implementasi *smart contract* pada aplikasi *e-voting*. Implementasi *smart contract* pada *e-voting* dengan metode *peer-to-peer blockchain* Ethereum [4] lebih berfokus pada perancangan sistem *e-voting* berbasis *blockckain*. Pada penelitian Implementasi Teknologi *Blockchain* Pada Aplikasi *E-Voting* Berbasis Web [7] menggunakan *smart contract blockchain* Ethereum lebih fokus pada implementasi teknologi *blockchain* pada *e-voting.* Pada penelitian ini, penulis akan menganalisa cara kerja *e-voting blockchain* sampai dapat memberikan tingkat keamanan yang tidak terbantahkan seperti *voter* yang sudah melakukan pemilihan tidak dapat lagi melakukan pemilihan dan blok yang telah ditambahkan tidak dapat diubah atau dihapus oleh siapapun sehingga suara yang masuk tidak bisa berkurang. Penelitian ini akan memberikan bukti empiris melalui pengujian properti spesifik menggunakan *fuzzer* Echidna yang merupakan pendekatan analisis keamanan yang lebih sistematik dan mendalam dibandingkan sekadar implementasi fungsional pada penelitian sebelumnya. Penelitian ini menggunakan *fuzzer* Echidna dikarenakan Echidna bekerja dengan cara "menyerang" kontrak yang telah dibuat secara intensif dengan ribuan transaksi acak, mencoba sekuat tenaga untuk merusak salah satu aturan yang telah ditetapkan yang sulit dilakukan dan membutuhkan waktu yang lama apabila dilakukan secara manual. Echidna juga dikenal cepat, efisien, dan mampu mensimulasikan berbagai variasi input, urutan eksekusi, serta pemanggilan fungsi yang agresif sehingga lebih mampu mengungkap bug logika yang halus namun kritis.

METODOLOGI PENELITIAN

a. Metodologi Penelitian

Dalam pelaksanaan penelitian ini, diterapkan perancangan metode terlebih dahulu untuk meminimalisir kesalahan dalam proses penelitian. Tahapan metodologi penelitian dapat dilihat pada Gambar 1.

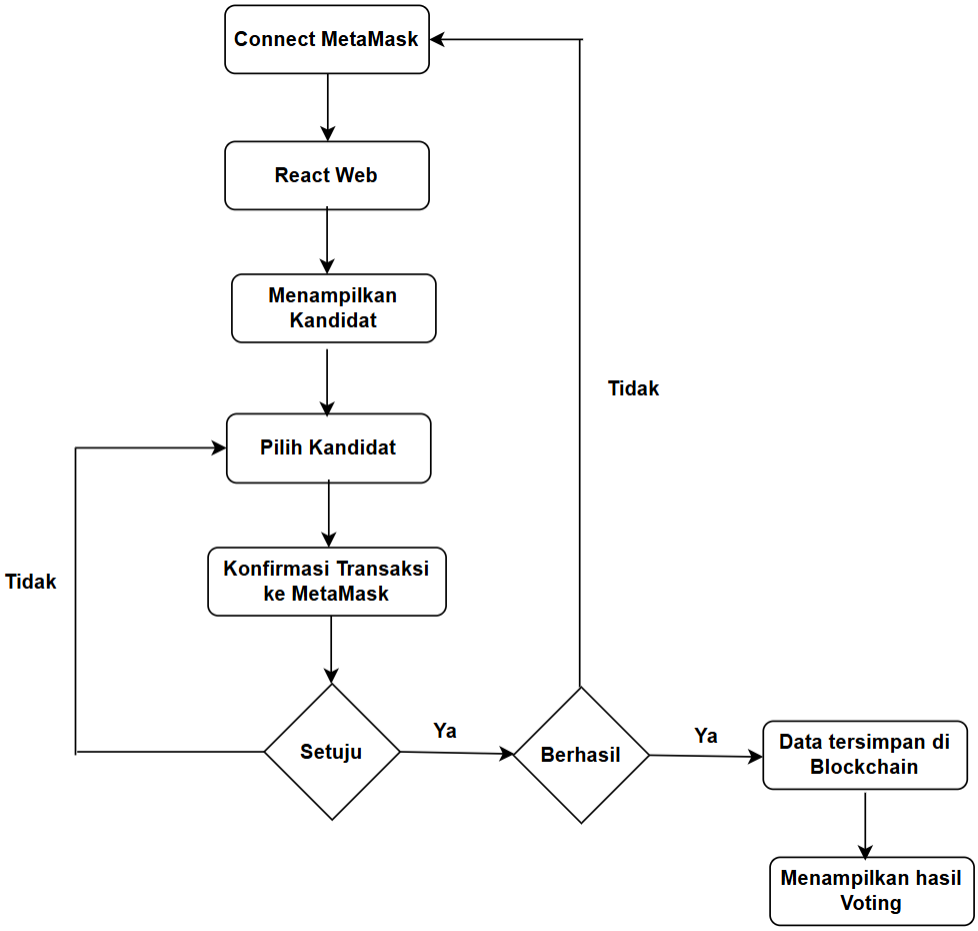


Gambar 1. Tahapan Metodologi Penelitian

1. Identifikasi Masalah
   Pada tahap ini dilakukan identifikasi masalah yang ada dari sistem *e-voting*, dalam hal ini dari segi keamanan sistem *e-voting* karena keamanan adalah masalah yang sangat penting dalam sistem *e-voting*. Adapun solusi yang dapat digunakan untuk mengatasi masalah keamanan dalam sistem *e-voting* adalah dengan menerapkan teknologi *blockchain*.
2. Studi Literatur
   Tahapan ini dimulai dengan melakukan pembelajaran literatur berupa jurnal dan buku elektronik terkait topik penelitian ini, yaitu analisis keamanan sistem *e-voting* berbasis *blokchain*.
3. Perancangan
   Pada tahapan ini dilakukan perancangan sistem *e-voting* yang dibutuhkan. Diawali dengan pembuatan *smart contract* dengan bahasa pemrograman Solidity. Dalam *smart contract* ditulis ketentuan pemilihan di mana satu alamat Ethereum hanya bisa melakukan 1 kali pemilihan dan hasil pemilihan akan disimpan dalam *blockchain*. Untuk *front end* nya menggunakan Reactjs yang ditulis dengan bahasa pemrograman JavaScript. Rancangan aplikasi voting ini menggunakan jaringan Ganache di mana jaringan ini menyediakan 50 *address* dengan 100 ETH

dummy di setiap *address* nya. Pada saat pemilihan diperlukan konfirmasi dari MetaMask *wallet*. Agar dapat menghubungkan Ganache dengan MetaMask *wallet*, jaringan MetaMask harus diubah ke jaringan lokal sesuai dengan alamat pada Ganache. Sebelum pemilihan, *address* pada Ganache harus di *import* ke MetaMask dengan menggunakan *private key* dari *address* yang ingin di *import*.

Smart contract yang telah dibuat harus di *compile* kemudian *migration* menggunakan JavaScript untuk mendapatkan file json dari file *contract*. Setelah berhasil, *script* Reactjs dijalankan agar *react web* muncul di *browser* untuk dapat melakukan pemilihan. Pada saat pemilihan akan ada konfirmasi dari MetaMask *wallet* yang membuktikan *address* tersebut sudah melakukan pemilihan. Setelah terkonfirmasi, suara akan masuk ke *blockchain*. Skema perancangan dapat dilihat pada Gambar 2.

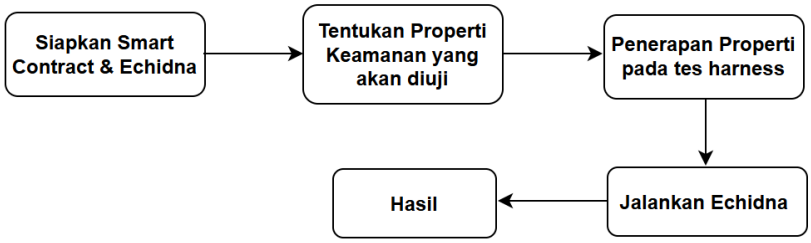


Gambar 2. Skema Perancangan *E-Voting*

4. Implementasi
   Pada tahapan ini dilakukan implementasi dari sistem yang telah dirancang sebelumnya agar sistem dapat melakukan pemilihan kandidat dan memastikan satu *address* hanya dapat melakukan satu kali pemilihan. Setelah pemilihan dilakukan, harus memastikan suara masuk ke *blockhain* agar suara yang masuk tidak dapat diubah atau dihapus.
5. Pengujian
   Pada tahapan ini, sistem yang telah dibuat sebelumnya akan dilakukan uji coba terhadap tingkat keamanan aplikasi dengan menggunakan Echidna yang meliputi lima properti yaitu tidak bisa *double vote*, total suara tidak pernah berkurang, jumlah kandidat selalu tetap, jumlah suara dan pemilih sama dan pemilih selalu memberikan satu suara. Kami melakukan pengujian lima

properti ini karena merupakan pilar integritas dari semua sistem pemungutan suara. Jika salah satu dari pilar ini runtuh, seluruh proses pemilu tidak dapat dipercaya.

Untuk melakukan pengujian menggunakan Echidna ada beberapa langkah yang harus dilakukan seperti mempersiapkan *smart contract* dan Echidna, menentukan properti keamanan yang akan diuji, menerapkan properti pada *tes harness* dan setelah itu Echidna dapat dijalankan. *Test harness* adalah kode pengujian yang menghubungkan sistem yang diuji dengan alat pengujian seperti Echidna. *Test harness* biasanya berupa kontrak Solidity tambahan yang harus dibuat pada folder yang sama dengan *smart contract*. Langkah-langkah untuk menjalankan Echidna dapat dilihat pada Gambar 3.



Gambar 3. Tahapan Pengujian Echidna

b. Rancangan Pengujian Keamanan

Ada beberapa tahapan penting yang dilakukan dalam penelitian ini agar dapat dilakukan analisis keamanan terhadap sistem *e-voting* berbasis *blockchain* seperti yang dapat dilihat pada Gambar 4.

Gambar 4. Rancangan Pengujian Keamanan

1. Perancangan *Smart Contract*
   Tahap ini merupakan tahap awal penelitian, yaitu membuat kontrak pintar yang berisi fungsi-fungsi utama dalam sistem *e-voting* yang meliputi daftar kandidat, menghitung jumlah suara, menyimpan status pemilih dan menambahkan suara pada kandidat yang dipilih.
2. Penentuan Properti Keamanan
   Untuk mencegah adanya bug pada *smart contract* sehingga dapat mempengaruhi hasil voting, maka akan dilakukan pengujian keamanan *smart contract* menggunakan Echidna. Properti yang ditetapkan dalam penelitian ini meliputi lima properti yaitu tidak bisa *double vote*, total suara tidak pernah berkurang, jumlah kandidat selalu tetap, jumlah suara dan pemilih sama dan pemilih selalu memberikan satu suara. Pemilihan lima properti ini dilakukan karena merupakan pilar integritas dari semua sistem pemungutan suara. Jika salah satu dari pilar ini runtuh, seluruh proses pemilu tidak dapat dipercaya.
3. Implementasi *Smart Contract* ke Ganache
   Smart contract yang telah dirancang dikompilasi dan dideploy menggunakan Truffle ke jaringan lokal Ganache. Proses ini bertujuan untuk menguji fungsi dasar dan memastikan bahwa sistem berjalan sesuai logika yang diharapkan sebelum dilakukan pengujian keamanan menggunakan Echidna.

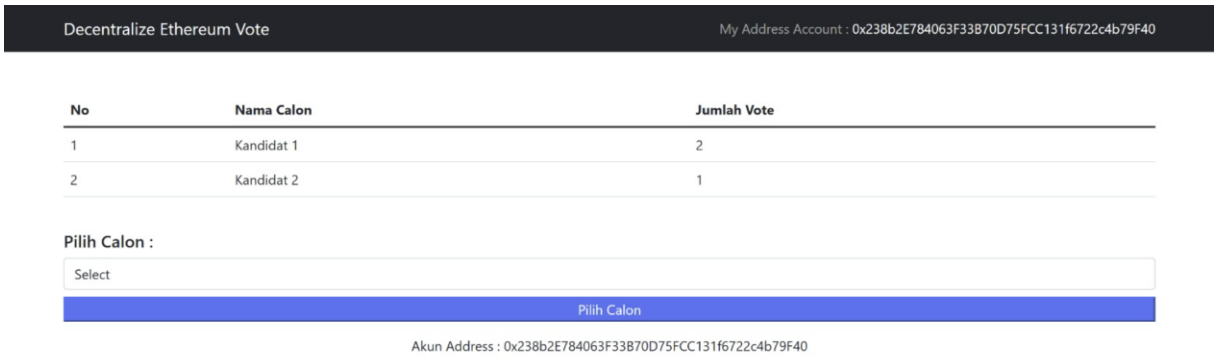
4. Pengujian Echidna
   Pengujian dilakukan dengan menulis kode Solidity tambahan yang harus dibuat pada folder yang sama dengan *smart contract* yang berisi definisi dari setiap properti yang telah ditetapkan. Echidna akan menjalankan *fuzzing test*, yaitu menguji berbagai input secara acak untuk menemukan kondisi yang dapat menyebabkan pelanggaran properti.
5. Analisis Hasil Pengujian
   Setelah pengujian selesai, data hasil uji Echidna dapat dianalisis untuk menilai sejauh mana *smart contract* memenuhi kriteria keamanan e-voting. Apabila hasil pengujian melaporkan *passing*, berarti kontrak tidak dapat dilanggar atau aman. Sebaliknya, jika hasil pengujian melaporkan *failed*, berarti ada bug yang dapat melanggar kontrak, sehingga kontrak masih ada celah dan perlu diperbaiki.

HASIL DAN PEMBAHASAN

Pada penelitian ini terdapat dua tahapan utama yaitu merancang sistem *e-voting* berbasis *blockchain* dan menganalisis sistem keamanan *e-voting* menggunakan *fuzzer* Echidna.

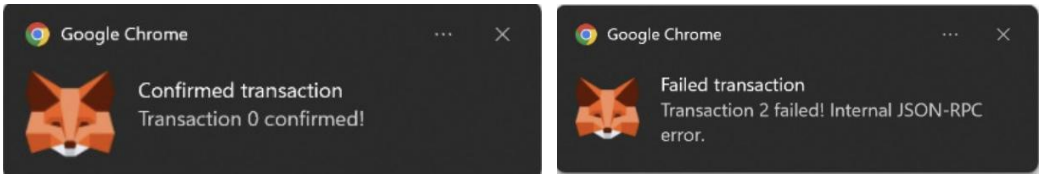
a. Perancangan Sistem *E-Voting*

Perancangan sistem e-voting ini terbagi menjadi dua bagian yaitu *frontend* dan *blockchain*. *Frontend* adalah halaman antarmuka yang digunakan oleh voter untuk melakukan pemilihan. Untuk *front end* nya menggunakan Reactjs yang ditulis dengan bahasa pemrograman JavaScript. Halaman antarmuka dari sistem e-voting ini dapat dilihat pada Gambar 5.



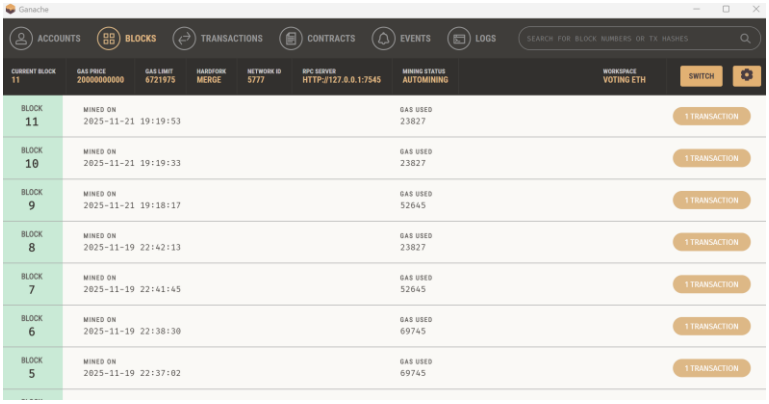
Gambar 5. Halaman Antarmuka *E-Voting*

Setiap kali pemilihan, akan ada konfirmasi dari *wallet* MetaMask. MetaMask adalah sebuah ekstensi browser yang populer dengan fungsi sebagai dompet *crypto* yang terhubung ke jaringan *blockchain* Ethereum. MetaMask telah menjadi pintu gerbang utama bagi jutaan pengguna dalam mengakses aplikasi desentralisasi (dApp), keuangan terdesentralisasi (DeFi), dan token NFT [14]. Jika voter belum pernah melakukan pemilihan maka akan muncul notifikasi konfirmasi transaksi dari MetaMask dan suara akan tercatat di *blockchain*. Tetapi apabila voter sudah pernah melakukan pemilihan dan akan melakukan pemilihan lagi maka akan muncul notifikasi transaksi gagal dari MetaMask dan suara tidak akan tercatat di blockchain walaupun pemilihan dapat dilakukan. Notifikasi transaksi berhasil dan transaksi gagal dapat dilihat pada Gambar 6.

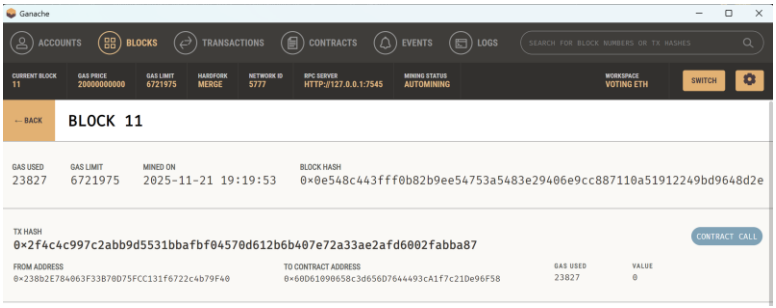


Gambar 6. Notifikasi Transaksi Berhasil dan Gagal

Suara dari voter akan di simpan di *blockchain*. *Blockchain* di sini bertindak sebagai *database* terdesentralisasi sehingga suara yang masuk tidak dapat diubah maupun dihapus. Rancangan aplikasi voting ini menggunakan jaringan Ganache di mana jaringan ini menyediakan 50 *address* dengan 100 ETH *dummy* di setiap *address* nya. Ganache adalah alat *blockchain* Ethereum untuk pribadi yang kuat dan dirancang efisien untuk mengimplementasikan dan mengamankan penyimpanan data. Alat ini menyediakan fitur-fitur yang membantu pengembang untuk menjalankan sistem *blockchain* Ethereum secara lokal di komputernya sendiri. Lingkungan Ganache dapat disesuaikan untuk membangun pengujian dan menerapkan *smart contract* serta dApps pada jaringan *blockchain* pribadi termasuk e-voting [15]. Pembuatan blok atas suara dari voter di *blockchain* dapat dilihat pada Gambar 7 dan informasi yang diberikan setiap blok dapat dilihat pada Gambar 8.



Gambar 7. Pembuatan Blok di *Blockchain*



Gambar 8. Informasi Blok di *Blockchain*

Suara yang sudah tercatat di blockchain tidak dapat diubah atau dihapus karena *blockchain* dikelola oleh jaringan komputer yang dikenal sebagai *node*, yang bekerja sama untuk memvalidasi dan mencatat transaksi. Keputusan apakah sebuah transaksi valid atau tidak, ditentukan oleh algoritma konsensus yang dijalankan oleh semua *node* yang terhubung di dalam jaringan *blockchain*. *Blockchain* terdiri dari serangkaian blok yang masing-masing berisi daftar transaksi. Setiap blok dalam *blockchain* ditautkan ke blok sebelumnya, yang keseluruhannya akan membuat rantai blok yang tidak dapat diubah atau dirusak [16]. Detail cara kerja blockchain yang membuatnya aman untuk diterapkan dalam sistem e-voting adalah:

1. Saat sebuah transaksi dilaksanakan dengan teknologi *blockchain* dalam hal ini voting, maka transaksi itu disebarluaskan ke semua *node* di dalam jaringan.
2. *Node* kemudian menverifikasi transaksi untuk memastikan bahwa transaksi itu sah atau valid dengan menggunakan seperangkat aturan yang dikodekan ke dalam protokol *blockchain* tersebut. Setelah transaksi diverifikasi, maka blok tersebut ditambahkan ke blok bersama dengan transaksi lain yang telah terjadi sejak blok terakhir dibuat.
3. Blok tersebut kemudian dienkripsi atau *hashed* dengan menggunakan teknik kriptografi, yang selanjutnya menciptakan sidik jari digital unik yang mewakili isi blok. *Hash* ini kemudian

ditambahkan ke rangkaian rantai blok sebelumnya yang selanjutnya akan membuat tautan antara dua blok dan memastikan bahwa isi blok tidak dapat diubah.
4. Setelah blok baru ditambahkan, seluruh jaringan akan memperbaharui salinan buku besar (*ledger*) mereka. Semua *node* dalam jaringan memiliki salinan catatan *blockchain* yang identik, memastikan data tersebut terdistribusi dan sulit untuk diubah atau dimanipulasi oleh satu pihak.
5. Transaksi tersebut kemudian dianggap selesai dan dicatat secara permanen dalam buku besar yang terdistribusi.

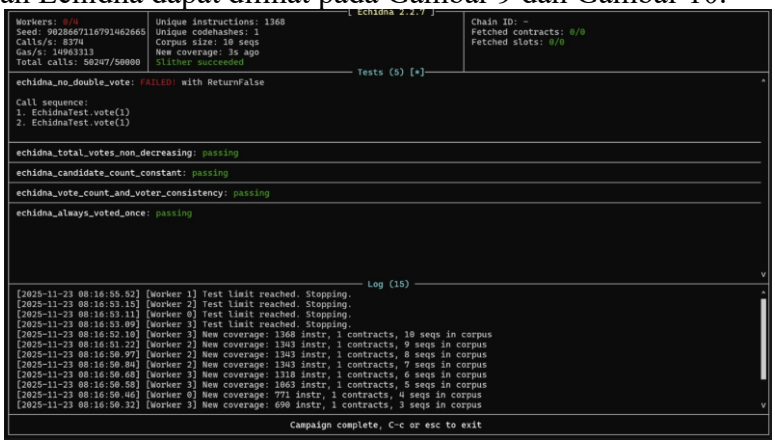
b. Pengujian Keamanan Menggunakan Echidna

Untuk memastikan sistem aman, maka dilakukan pengujian menggunakan *fuzzer* Echidna. Echidna yaitu sebuah *fuzzer* berbasis properti yang menguji kerentanan keamanan pada kontrak pintar Ethereum. Echidna bekerja dengan cara "menyerang" kontrak yang telah dibuat secara intensif dengan ribuan transaksi acak, mencoba sekuat tenaga untuk merusak salah satu aturan yang telah ditetapkan. Echidna digunakan oleh pengembang kontrak maupun auditor internal di *Trail of Bits* karena Echidna adalah *fuzzer* paling populer dan paling terdukung untuk kontrak pintar Ethereum [17]. Echidna dapat menggunakan beberapa metode pengujian yaitu [17] :

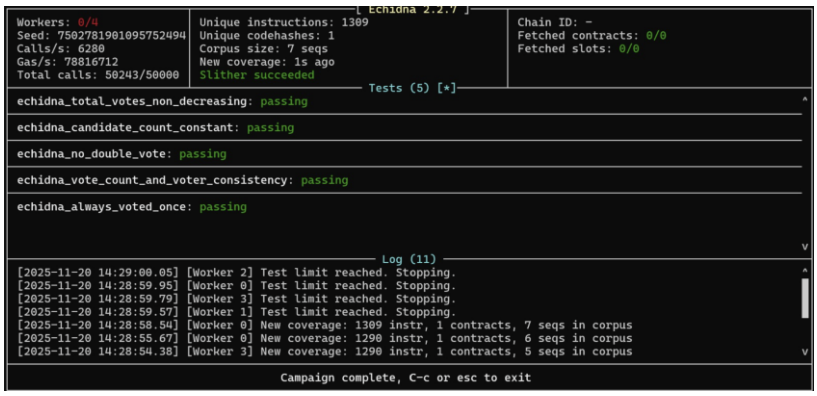
1) *Swarm Testing (API Call Omission)*
Swarm adalah infrastruktur penyimpanan dan komunikasi terdesentralisasi yang dirancang untuk mengatasi kebutuhan yang semakin meningkat akan privasi, keamanan, dan efisiensi dalam pengelolaan data. Arsitektur yang tangguh dan penggunaan teknologi *blockchain* yang inovatif membuatnya ideal untuk berbagai kasus penggunaan termasuk sistem pemungutan suara berbasis *blockchain*. Karena *Swarm* menyediakan solusi penyimpanan yang aman dan tahan sensor untuk aplikasi terdesentralisasi (dApps) dalam ekosistem Web3.
2) *Test Length Variance*
Echidna tidak hanya menguji satu transaksi tunggal, tetapi juga kombinasi dari beberapa transaksi, kadang urutannya panjang (misalnya 10 transaksi), kadang pendek (1–2 transaksi). Tujuannya adalah untuk mengeksplorasi lebih luas *state space* kontrak pintar, karena bug sering muncul bukan karena satu transaksi saja, tetapi karena urutan transaksi tertentu. Jika dikaitkan dengan *e-voting*, Echidna menguji apakah kontrak tetap aman saat urutan transaksi bertambah panjang (misal banyak *voter* melakukan aksi berurutan).
3) *Mutation and Search Variance*
Dalam Echidna, *mutation* berarti perubahan acak terhadap input transaksi yang diuji. Echidna menggunakan strategi *fuzzing* berbasis *mutase* dengan membuat banyak versi baru dari urutan transaksi dengan mengubah urutan fungsi yang dipanggil dan parameter input (misalnya alamat *voter*, ID kandidat). Echidna menguji berbagai kombinasi *voter*, kandidat, dan urutan aksi untuk menemukan bug seperti *double vote* dan *invalid candidate*.
Search Variance mengukur keragaman jalur eksekusi *(execution paths)* yang dijelajahi oleh Echidna selama proses *fuzzing*. Setiap kali kontrak mengeksekusi cabang logika berbeda (misalnya *if–else* atau *require* yang berbeda), itu dihitung sebagai jalur baru. Echidna mengeksplorasi semua cabang logika voting dan validasi karena menjamin semua kondisi logika diuji dan tidak ada celah tersembunyi.
4) *User-Controlled Variance*
User-Controlled Variance adalah ukuran seberapa besar pengaruh variasi input dari pengguna terhadap perilaku kontrak pintar selama *fuzz testing*. Echidna mengukur dampak variasi *voter*, kandidat, urutan voting agar potensi bug input ditemukan lebih banyak.

Pengujian Echidna pada penelitian ini menggunakan Solidity versi 0.8.0 dan dijalankan menggunakan Docker. Proses pengujian membutuhkan waktu lebih kurang 8 detik dengan rata-rata 50.000 *calls*, dengan kecepatan pemanggilan fungsi sekitar 6.280 *calls* per *second*. Total *calls* dapat melebihi 50.000 *calls* tergantung berat ringannya kontrak yang diuji dan jalur eksekusi yang sedang

dilakukan optimal. *Calls* adalah jumlah panggilan fungsi yang dilakukan Echidna terhadap *smart contract* selama proses pengujian. Pengujian Echidna pada penelitian ini menggunakan lima properti keamanan *smart contract* yaitu tidak bisa *double vote*, total suara tidak pernah berkurang, jumlah kandidat selalu tetap, jumlah suara dan pemilih sama dan pemilih selalu memberikan satu suara. Hasil pengujian Echidna dapat dilihat pada Gambar 9 dan Gambar 10.



Gambar 9. Pengujian Echidna yang Menghasilkan *Failled*



Gambar 10. Pengujian Echidna yang Menghasilkan *Passing*

Pada Gambar 9 di atas dapat dilihat bahwa properti pertama yaitu tidak bisa *double vote* gagal pada pengujian Echidna. Ini menandakan *smart contract* yang telah dibuat ada potensi untuk di langgar sehingga perlu dilakukan perbaikan *smart contract* agar sistem yang dibuat aman dari *bug*. Pada Gambar 10 dapat dilihat bahwa semua properti yang diuji telah berhasil yang berarti *smart contract* yang dibuat aman dari bug karena Echidna bekerja secara *fuzzing* dengan ribuan transaksi acak sehingga dapat memastikan sistem yang dibuat aman. Penjelasan dari hasil pegujian dapat dilihat pada Tabel 1 di bawah ini.

Tabel 1. Penjelasan Hasil Pengujian

Properti	Status	Penjelasan
Tidak bisa *double vote*	*Passing*	Tidak ditemukan pelanggaran, artinya 1 akun hanya dapat melakukan 1 kali voting
Total suara tidak pernah berkurang	*Passing*	Nilai vote selalu meningkat dan sistem bebas dari manipulasi pengurangan suara
Jumlah kandidat selalu tetap	*Passing*	Jumlah kandidat yang telah ditetapkan tidak dapat diubah
Jumlah suara dan pemilih sama	*Passing*	Jumlah pemilih yang telah melakukan vote selalu sama dengan jumlah suara yang masuk

Properti	Status	Penjelasan
Pemilih selalu memberikan satu suara	*Passing*	Pada saat vote, setiap pemilih hanya dapat memberikan satu suara

Berdasarkan penjelasan hasil pengujian pada Tabel 1 dimana semua properti yang ditentukan telah lolos dari pengujian menunjukkan bahwa *smart contract* dari *e-voting* yang dikembangkan berjalan dengan baik dan dapat menjamin keamanan sistem *e-voting*. Oleh karena itu implementasi *smart contract* pada *blockchain* Ethereum berpotensi untuk diaplikasikan pada sistem pemilihan *e-voting* karena dapat menjaga keamanan dan transparansi pemilihan. Selain itu pengujian menggunakan sistem *fuzzing* terbukti penting dilakukan untuk memastikan keamanan *smart contract* sebelum diimplementasikan secara nyata.

KESIMPULAN

Berdasarkan hasil pengujian *smart contract* pada sistem *e-voting* berbasis *blockchain* dengan menggunakan *fuzzer* Echidna, dapat disimpulkan bahwa semua properti yang ditentukan yaitu tidak bisa *double vote*, total suara tidak pernah berkurang, jumlah kandidat selalu tetap, jumlah suara dan pemilih sama dan pemilih selalu memberikan satu suara memberikan hasil *passing* walaupun telah diuji dengan ribuan transaksi acak yang artinya *smart contract* aman dan dapat diterapkan secara nyata. Hal ini membuktikan bahwa *smart contract* telah dirancang secara baik dan sesuai yang diharapkan.

Penelitian ini dapat memberikan bukti empiris bahwa pengujian menggunakan *fuzzer* berbasis properti seperti Echidna dapat membantu mendeteksi kerentanan logika pada *smart contract* sebelum diterapkan ke jaringan *blockchain* nyata. Selain itu, pendekatan ini dapat menjadi referensi bagi pengembang sistem *e-voting* serta peneliti keamanan dalam membangun mekanisme pemungutan suara yang lebih transparan dan tahan terhadap manipulasi.

Namun penelitian ini masih memiliki keterbatasan dimana Echidna hanya melakukan pengujian berdasarkan properti yang telah ditentukan dan juga penelitian ini menggunakan jaringan lokal Ganache yang belum bisa mencerminkan kondisi jaringan *blockchain* publik yang sesungguhnya. Oleh karena itu untuk penelitian selanjutnya disarankan untuk dapat dilakukan penambahan properti seperti verifikasi hasil akhir pemilihan dan juga dapat melakukan pengujian keamanan saat diimplementasikan pada jaringan *blockchain* publik agar mendapat gambaran nyata.

DAFTAR PUSTAKA

[1] H. Saputro, "Sistem Informasi E-Voting Dengan Metode Rapid Application Development (Rad) Pada Pemilihan Kepala Desa Berbasis Website," *Biner J. Ilm. Inform. dan Komput.*, vol. 1, no. 1, pp. 43–51, 2022, doi: 10.32699/biner.v1i1.2500.

[2] Junaedi, A. Fernando, and A. Hermawan, "Aplikasi E-voting Berbasis Blockchain dengan Metode Smart Contract," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 6, no. 2, pp. 408–415, 2024.

[3] J. F. Potalangi, D. P. Kartikasari, and N. H. Shaffan, "Implementasi Jaringan Permissioned Blockchain pada Sistem E-Voting Pemilwa untuk Menjamin Autentikasi Pemilih dan Integritas Data," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 4, pp. 2548–964, 2025, [Online]. Available: http://j-ptiik.ub.ac.id

[4] G. P. Putra, "Implementasi Smart Contract Pada E-Voting Dengan Metode Peer-To-Peer Blockchain Ethereum," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 12, no. 1, pp. 118–125, 2025, doi: 10.30656/prosisko.v12i1.9136.

[5] R. Adadi Suparlan, "Implementasi Smart Contract Blockchain Ethereum Pada Aplikasi E-voting," *Informatics Digit. Expert*, vol. 7, no. 1, pp. 66–70, 2025, doi: 10.36423/index.v7i1.2177.

[6] H. O. Ohize *et al.*, *Blockchain for securing electronic voting systems: a survey of architectures, trends, solutions, and challenges*, vol. 28, no. 2. Springer US, 2025. doi: 10.1007/s10586-024-04709-8.
[7] A. Ilahi, D. Kurniadi, D. Novaliendry, and T. Sriwahyuni, "Implementasi Teknologi Blockchain Pada Aplikasi E-Voting Berbasis WEB," *J. Pendidik. Tambusai*, vol. 8, pp. 28924–28938, 2024, [Online]. Available: https://jptam.org/index.php/jptam/article/download/18602/13884/34260?
[8] A. Nanda Sari and T. Gelar, "Blockchain: Teknologi Dan Implementasinya," *J. Mnemon.*, vol. 7, no. 1, pp. 63–70, 2024, doi: 10.36040/mnemonic.v7i1.6961.
[9] B. Raharjo, *Uang Masa depan*, no. 73. 2022.
[10] A. Razaque *et al.*, "Blockchain-Enabled Smart Contracts and Prioritized Delegated Proof-of-Stake Paradigm for Secure and Scalable Electronic Voting Systems," *Blockchain Res. Appl.*, p. 100348, 2025, doi: 10.1016/j.bcra.2025.100348.
[11] R. Suwardiyati, H. N. Widhiyanti, and S. Wicaksono, "Sah atau Tidak Smart Contract Dalam Sistem Blockchain?," *Widya Yuridika*, vol. 7, no. 2, pp. 459–468, 2024, doi: 10.31328/wy.v7i2.5156.
[12] G. L. Saroinsong, A. Sambul, and S. R. U. A. Sompie, "Implementasi of Blockchain Technology in E-Voting System," *Tek. Inform.*, vol. 20, no. 1, pp. 11–18, 2025.
[13] P. Teknologi, B. Pada, G. L. Saroinsong, A. Sambul, and S. R. U. A. Sompie, "E-Voting Systems," vol. 20, no. 1, pp. 11–18, 2025.
[14] F. Kalvin, M. I. Sa'ad, and A. F. Pukeng, "Implementasi Seed Phrase Dalam Keamanan Dompet Kripto Pada Metamask," *Bull. Inf. Technol.*, vol. 6, no. 2, pp. 136–147, 2025, doi: 10.47065/bit.v5i2.2026.
[15] A. Fikri, M. F. Andrijasa, T. Bustomi, P. N. Samarinda, S. Keledang, and S. Seberang, "Implementasi Teknologi Smart Contracts Untuk Sistem Ijazah Digital Di Politeknik Negeri," vol. 3, no. 6, 2025.
[16] E. P. Hendraswara *et al.*, "Teknologi Blockchain Dan Potensi Pemanfaatnnya Di Indonesia 2023," *Pokja PANDI*, pp. 1–87, 2023.
[17] A. Groce and G. Grieco, "Echidna-parade: A tool for diverse multicore smart contract fuzzing," *ISSTA 2021 - Proc. 30th ACM SIGSOFT Int. Symp. Softw. Test. Anal.*, pp. 658–661, 2021, doi: 10.1145/3460319.3469076.