Jurnal Teknik Informatika dan Teknologi Informasi
Volume. 5 Nomor. 2 Agustus 2025
E-ISSN: 2827-9387, P- ISSN: 2827-9379, Hal 486-498
DOI: https://doi.org/10.55606/jutiti.v5i2.5665
Available Online at: https://journalshub.org/index.php/jutiti

Pengembangan Prototipe Token Transaksi Cryptocurrency SDSPay Berbasis Blockchain Ethereum

Virgian Galang Sasongko[1], Faulinda Ely Nastiti[2], Sopingi[3]
[1-3] Universitas Duta Bangsa Surakarta, Indonesia
Email: virgiangalang@gmail.com[1], faulinda_ely@udb.ac.id[2], sopingi@udb.ac.id[3]

Abstract: *The transformation of digital payment systems through blockchain technology has brought new challenges and opportunities in the development of secure, efficient, and decentralized crypto tokens. One emerging approach is the use of smart contract-based tokens, such as ERC-20 tokens running on the Ethereum network. This research proposes the design and implementation of the SDSPay (SDS) token as a prototype Ethereum-based ERC-20 token, with a modern smart contract approach to meet the needs of a more efficient and secure digital payment system. The SDS token adopts the basic ERC-20 standard but is equipped with advanced features that enhance functionality and security, such as role-based access control (RBAC) to regulate access and control over transactions, pauseable transactions to pause transactions if necessary, and compatibility with EIP-2612 permits that enable more efficient transaction authorization in terms of gas. These features are designed to improve the efficiency and security of transactions on blockchain networks, thus enabling the use of tokens in a more reliable digital payment system. The SDS token prototype was tested on the Sepolia Testnet using Remix IDE and MetaMask to develop and manage smart contracts. Additionally, a static security audit was conducted using Slither Analyzer to detect potential vulnerabilities. The test results showed that the SDS token was successfully deployed and performed well, with an average transaction time of 10–12 seconds and stable gas fees. The Slither audit also found no significant vulnerabilities, indicating that the smart contract structure adheres to security best practices. This study confirms that the development of standardized smart contract-based tokens can be carried out using an efficient, reliable, and replicable methodology for other applications in future blockchain-based payment systems. This implementation of the SDSPay (SDS) token can serve as a foundation for designing secure and efficient digital payment systems, paving the way for the broader development of blockchain technology.*

Keywords: *Blockchain, ERC-20, Ethereum, SDS Token, Smart Contracts*

Abstrak: Transformasi sistem pembayaran digital melalui teknologi blockchain telah membawa tantangan sekaligus peluang baru dalam pengembangan token kripto yang aman, efisien, dan terdesentralisasi. Salah satu pendekatan yang berkembang adalah penggunaan token berbasis kontrak pintar, seperti token ERC-20 yang berjalan pada jaringan Ethereum. Penelitian ini mengusulkan perancangan dan implementasi token SDSPay (SDS) sebagai prototipe token ERC-20 berbasis Ethereum, dengan pendekatan kontrak pintar modern untuk memenuhi kebutuhan sistem pembayaran digital yang lebih efisien dan aman. Token SDS mengadopsi standar ERC-20 dasar namun dilengkapi dengan fitur-fitur lanjutan yang meningkatkan fungsionalitas dan keamanan, seperti role-based access control (RBAC) untuk mengatur akses dan kontrol atas transaksi, pausable transaction untuk menghentikan sementara transaksi jika diperlukan, serta kompatibilitas dengan EIP-2612 permit yang memungkinkan otorisasi transaksi yang lebih efisien dari segi gas. Fitur-fitur ini dirancang untuk meningkatkan efisiensi dan keamanan transaksi pada jaringan blockchain, sehingga memungkinkan penggunaan token dalam sistem pembayaran digital yang lebih handal. Prototipe token SDS diuji pada Sepolia Testnet dengan menggunakan Remix IDE dan MetaMask untuk mengembangkan dan mengelola kontrak pintar. Selain itu, audit keamanan statis dilakukan menggunakan Slither Analyzer untuk mendeteksi potensi kerentanannya. Hasil pengujian menunjukkan bahwa token SDS berhasil dideploy dan dijalankan dengan baik, dengan waktu transaksi rata-rata 10–12 detik dan biaya gas yang stabil. Audit Slither juga tidak menemukan kerentanannya yang signifikan, yang menunjukkan bahwa struktur kontrak pintar ini telah mengikuti praktik terbaik dalam pengamanan. Studi ini menegaskan bahwa pengembangan token berbasis kontrak pintar yang terstandarisasi dapat dilakukan dengan metodologi yang efisien, dapat diandalkan, dan dapat direplikasi untuk aplikasi lain dalam sistem pembayaran berbasis blockchain di masa depan. Implementasi token SDSPay (SDS) ini dapat menjadi landasan dalam merancang sistem pembayaran digital yang aman dan efisien, membuka jalan bagi pengembangan teknologi blockchain yang lebih luas.

Kata kunci: Blockchain, ERC-20, Ethereum, Kontrak Cerdas, Token SDS

Received: Juli 06, 2025; Revised: Juli 20, 2025; Accepted: Agutsu 03, 2025; Published: Agustus 05, 2025

1. PENDAHULUAN

Latar Belakang

SDS IT Consulting adalah perusahaan penyedia layanan teknologi informasi yang berfokus pada solusi data analytics dan pengembangan sistem digital untuk klien nasional maupun internasional. Dalam praktik bisnisnya, perusahaan menghadapi tantangan signifikan dalam transaksi keuangan lintas negara, khususnya dalam aspek efisiensi, transparansi, dan keamanan. Kompleksitas ini semakin tinggi akibat kebutuhan integrasi antara sistem pembayaran internal dengan berbagai platform eksternal milik klien luar negeri, yang melibatkan berbagai mata uang serta sistem perbankan konvensional [1].

Model pembayaran konvensional yang masih mengandalkan perantara seperti bank atau penyedia jasa transfer uang tradisional memiliki sejumlah keterbatasan, antara lain waktu pemrosesan yang lama, biaya transfer tinggi, serta potensi keterlambatan karena perbedaan zona waktu dan regulasi internasional. Tantangan ini menjadi pendorong utama untuk merancang sistem pembayaran alternatif yang mampu menyederhanakan proses transaksi lintas negara tanpa mengorbankan aspek keamanan, efisiensi, dan akuntabilitas [1], [2], [3].

Dalam konteks tersebut, **teknologi blockchain** hadir sebagai pendekatan yang menjanjikan. Melalui mekanisme desentralisasi dan sistem pencatatan yang immutable, blockchain memungkinkan pelaksanaan transaksi secara peer-to-peer tanpa keterlibatan otoritas sentral [4], [5] . Salah satu implementasi teknis yang paling banyak digunakan dalam transaksi digital adalah token berbasis smart contract, terutama yang mengikuti standar ERC-20 di jaringan Ethereum. Token jenis ini terbukti mampu mendukung proses transaksi yang transparan dan dapat diaudit [5].

Namun, implementasi token digital seperti ERC-20 tidak terlepas dari berbagai tantangan teknis. Di antaranya adalah tingginya biaya transaksi (*gas fee*), keterbatasan fleksibilitas pengaturan hak akses, serta belum optimalnya mekanisme pengendalian terhadap risiko-risiko darurat seperti serangan *reentrancy* dan *integer overflow*[4], [5], [6],[7]. Beberapa fitur lanjutan yang telah diusulkan dalam berbagai penelitian mencakup komponen *pausable*, *permit*, dan audit otomatis menggunakan alat bantu seperti Slither dan VisualGas untuk menjamin keamanan kontrak pintar sebelum diimplementasikan di jaringan publik [7], [8], [9], [10]

Berdasarkan tantangan-tantangan tersebut serta kebutuhan internal SDS IT Consulting, penelitian ini bertujuan untuk merancang dan mengembangkan prototipe token digital bernama **SDSPay**, yang dirancang sebagai solusi transaksi internal maupun lintas negara. SDSPay diharapkan mampu mengatasi berbagai kendala teknis dan keamanan yang sering muncul

dalam transaksi internasional, serta diujicobakan secara fungsional pada jaringan Ethereum testnet (Sepolia) sebagai bentuk validasi awal terhadap performa sistem [5], .

2. METODE PENELITIAN

Metode penelitian yang digunakan dalam studi ini adalah metode eksperimen terapan dengan pendekatan rekayasa perangkat lunak (software engineering-based experimental approach). Penelitian ini dilakukan secara sistematis melalui beberapa tahapan utama, yaitu: analisis kebutuhan sistem, perancangan kontrak pintar berbasis ERC-20, implementasi pada jaringan Ethereum testnet, pengujian fungsional, serta analisis keamanan dan efisiensi gas.

Pendekatan ini umum digunakan dalam pengembangan solusi teknologi berbasis blockchain, khususnya untuk implementasi dan evaluasi kontrak pintar pada jaringan Ethereum. Seperti yang dijelaskan oleh Signer, eksperimen terapan dalam konteks Ethereum memungkinkan identifikasi titik-titik konsumsi gas tertinggi dalam proses transaksi serta evaluasi efisiensi struktur kode [11]. Dengan demikian, pendekatan ini tidak hanya menekankan pada fungsionalitas sistem, tetapi juga pada aspek optimasi performa dan mitigasi risiko dalam arsitektur kontrak.

Alat dan Bahan

Dalam penelitian ini, sejumlah perangkat lunak dan teknologi pendukung digunakan untuk menunjang proses pengembangan, implementasi, dan pengujian prototipe token digital berbasis smart contract. Bahasa pemrograman yang digunakan adalah Solidity versi ^0.8.20, yakni bahasa utama untuk menulis kontrak pintar di jaringan Ethereum. Versi ini dipilih karena menawarkan sejumlah peningkatan fitur keamanan, termasuk penanganan overflow/underflow secara otomatis serta peningkatan efisiensi eksekusi instruksi.

Pengembangan kontrak dilakukan menggunakan **Remix IDE**, sebuah lingkungan pengembangan terintegrasi berbasis web yang mendukung kompilasi, deploy, serta pengujian kontrak pintar langsung di browser. Pemanfaatan Remix IDE memungkinkan iterasi cepat dalam proses pengembangan tanpa memerlukan setup lokal yang kompleks.

Untuk pengelolaan akun pengguna dan penyimpanan kunci privat, digunakan **MetaMask**, yaitu dompet kripto berbasis perangkat lunak yang tersedia dalam bentuk aplikasi iOS serta ekstensi browser (Chrome). MetaMask memungkinkan interaksi langsung antara pengguna dan jaringan blockchain Ethereum, termasuk proses deployment kontrak dan pengiriman transaksi.

Jaringan yang digunakan dalam pengujian adalah **Ethereum Sepolia Testnet**, yaitu jaringan uji coba resmi dari Ethereum yang mendukung simulasi deployment dan eksekusi

kontrak secara nyata tanpa menggunakan ether asli. Untuk dapat melakukan transaksi pada jaringan ini, peneliti memanfaatkan **token uji SepoliaETH** yang diperoleh secara gratis dari faucet publik.

Sebagai bagian dari pengujian keamanan, digunakan **Slither Static Analyzer**, sebuah alat analisis statis yang secara otomatis memindai kontrak pintar untuk mendeteksi potensi kerentanan seperti reentrancy, integer overflow, dan race condition. Slither dipilih karena telah diakui sebagai salah satu tools paling andal untuk analisis keamanan smart contract dalam industri blockchain modern.

Seluruh transaksi dan status kontrak yang diimplementasikan diverifikasi menggunakan **Etherscan Sepolia**, yaitu penjelajah blok (block explorer) yang menyediakan antarmuka publik untuk memantau aktivitas transaksi, status deployment kontrak, serta rincian data interaksi blockchain secara transparan. Penggunaan Etherscan menjadi bagian dari validasi publik bahwa sistem telah diuji secara nyata pada infrastruktur blockchain yang relevan.

Tahapan Penelitian

Proses penelitian ini mengikuti pendekatan pengembangan perangkat lunak secara adaptif yang terdiri dari lima tahapan utama: analisis kebutuhan dan studi literatur, perancangan kontrak pintar, implementasi, pengujian fungsional, serta audit keamanan. Setiap tahapan dirancang untuk membentuk kerangka kerja sistematis dalam pembangunan dan evaluasi token digital berbasis ERC-20.

- Analisis dan Studi Literatur

  Tahapan awal melibatkan pengumpulan dan telaah literatur terkait standar ERC-20, praktik terbaik desain modular, serta integrasi fitur keamanan lanjutan. Peninjauan mencakup fitur seperti pausable, role-based access control, dan permit sesuai standar EIP-2612. Studi terdahulu telah membahas dampak kontrak pintar terhadap arsitektur keuangan [9], serta menekankan pentingnya modularitas dan keamanan dalam desain kontrak pintar [9]**.**

- Perancangan Kontrak Pintar SDS

  Kontrak pintar dirancang menggunakan pustaka OpenZeppelin Contracts yang telah distandarisasi secara luas dalam komunitas Ethereum. Modul yang diintegrasikan meliputi ERC-20, Burnable, Pausable, Permit, dan AccessControl. Arsitektur yang diterapkan memisahkan logika inti (core logic) dari fitur lanjutan (extension), guna

mendukung modularitas dan kemudahan audit. Pendekatan ini telah direkomendasikan dalam berbagai penelitian sebagai strategi untuk meningkatkan fleksibilitas serta mengurangi risiko kerentanan akibat kompleksitas logika tunggal [8], [9] .

- Implementasi di Jaringan Ethereum Testnet

  Implementasi kontrak dilakukan melalui Remix IDE, dengan koneksi ke jaringan Ethereum Sepolia Testnet menggunakan MetaMask (Injected Provider). Dalam proses deployment, ditentukan sejumlah parameter penting seperti initialSupply, name, symbol, dan role akses pengguna. Untuk mendukung proses transaksi tanpa biaya aktual, digunakan token uji SepoliaETH yang diperoleh melalui faucet publik. Penggunaan testnet sebagai sarana simulasi dan validasi teknis merupakan praktik umum dalam pengembangan sistem berbasis blockchain [8].
- Pengujian Fungsional Token SDS

  Pengujian dilakukan terhadap fungsi utama token, meliputi mint, transfer, approve, serta fitur kontrol pause/unpause. Seluruh fungsi diuji dalam skenario transaksi antar dua akun MetaMask, dan setiap transaksi diverifikasi menggunakan Etherscan Sepolia. Selain itu, waktu transaksi dan konsumsi gas dicatat untuk mengukur efisiensi eksekusi kontrak. Pengukuran konsumsi gas dinilai sebagai aspek krusial dalam evaluasi kontrak Ethereum, terutama dalam konteks optimasi biaya [8].
- Audit Keamanan dengan Slither

  Audit keamanan dilakukan menggunakan Slither Static Analyzer, yaitu alat analisis statis yang secara otomatis mengidentifikasi potensi kerentanan dalam kode Solidity. Fokus audit mencakup celah umum seperti reentrancy, unused return, approve race condition, serta pelanggaran terhadap standar ERC-20. Teknik audit ini terbukti efektif dalam mengidentifikasi celah pada tahap awal pengembangan sebelum kontrak dirilis ke jaringan publik [8],[12].

3. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi kontrak pintar token **SDS**, serta pembahasan menyeluruh terkait aspek fungsionalitas, efisiensi transaksi, dan keamanan sistem. Seluruh pengujian dilakukan pada jaringan Ethereum testnet (Sepolia), sementara audit keamanan menggunakan metode analisis statis melalui **Slither Static Analyzer**. Setiap hasil yang disampaikan dalam bab ini mengikuti tahapan yang telah dijelaskan pada Bab 2.

Hasil Implementasi Token SDS

Implementasi kontrak pintar token **SDS** dilaksanakan berdasarkan tahapan analisis dan perancangan sistem sebagaimana dijelaskan pada Bab 2. Proses implementasi dilakukan menggunakan **Remix IDE** dengan integrasi dompet kripto **MetaMask** sebagai *injected provider* untuk koneksi ke jaringan **Ethereum Sepolia Testnet**. Kompilasi kontrak menggunakan **Solidity compiler versi 0.8.20**, menghasilkan *bytecode* dan ABI yang sesuai dengan spesifikasi kontrak ERC-20. Kontrak berhasil dideploy dan diverifikasi melalui **Etherscan Sepolia** dengan alamat kontrak sebagai berikut:
0x4A615396fB2183E978E3B9b85b070706B3f08420

Dalam proses perancangan, kontrak dibangun secara modular menggunakan pustaka **OpenZeppelin** yang mencakup komponen ERC20, Burnable, Pausable, AccessControl, serta Permit (EIP-2612). Struktur modular ini memisahkan logika dasar dari fitur lanjutan, sehingga memungkinkan fleksibilitas dalam pengelolaan hak akses dan fungsi tambahan yang dapat diaktifkan secara selektif. Arsitektur tersebut mendukung prinsip keamanan dan pemeliharaan yang baik dalam pengembangan kontrak pintar.

Parameter awal seperti initialSupply, name, symbol, dan pengaturan role ditentukan pada saat proses deployment. Setelah berhasil dideploy, token SDS terdeteksi secara otomatis pada antarmuka MetaMask, dan seluruh aktivitas transaksi dapat dimonitor melalui Etherscan, termasuk validasi terhadap event penting seperti Transfer, Paused, dan Unpaused.

Uji fungsional menunjukkan bahwa seluruh fitur kontrak telah berfungsi sebagaimana dirancang. Fungsi dasar seperti transfer, approve, dan transferFrom berjalan dengan baik, begitu pula fitur lanjutan mint, burn, pause, dan unpause. Implementasi kontrol akses berbasis peran juga berhasil diaktifkan, dengan definisi DEFAULT_ADMIN_ROLE, MINTER_ROLE, dan PAUSER_ROLE yang masing-masing dapat dibuktikan fungsionalitasnya melalui pengujian antar dua akun MetaMask.

Meskipun fungsi permit (EIP-2612) telah disertakan dalam struktur kontrak, fitur ini belum diaktifkan secara eksplisit karena belum dilakukan integrasi fungsi tanda tangan off-chain dalam pengujian tahap ini. Secara keseluruhan, hasil implementasi ini menunjukkan bahwa kontrak SDS telah berhasil dioperasikan sesuai dengan tahapan perancangan, implementasi, dan pengujian yang dirancang sebelumnya.

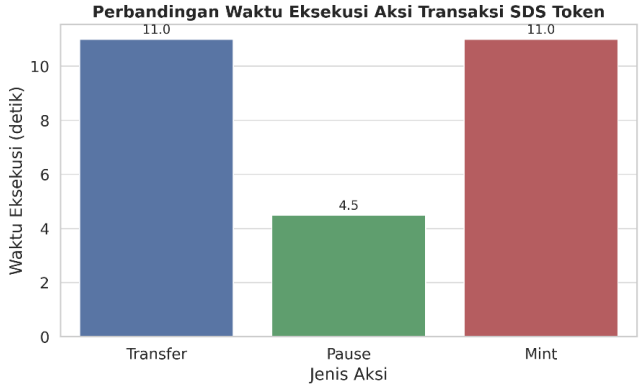
Pengujian Fungsional dan Efisiensi Transaksi

Pengujian dilakukan dengan mentransfer token antar dua akun MetaMask. Hasil transaksi diamati menggunakan Etherscan untuk mencatat waktu, biaya gas, dan status transaksi. Berikut adalah ringkasan hasil uji coba:

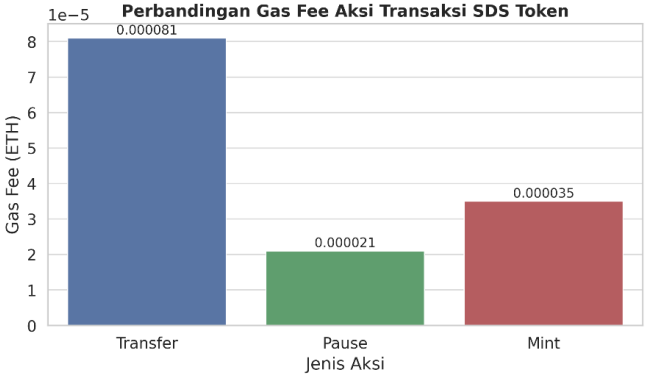
Tabel 1. Ringkasan Hasil Pengujian Fungsional Token SDS

No	Aksi	Jumlah Token	Waktu Eksekusi (detik)	Gas Fee (ETH)	Status
1	Transfer	1.000 SDS	11,0	0.000081	Berhasil
2	Pause Transaksi	-	4,5	0.000021	Berhasil
3	Mint Token	5.000 SDS	11,0	0.000035	Berhasil

Seluruh transaksi menggunakan saldo 0.05 SepoliaETH yang diperoleh dari faucet publik. Waktu eksekusi rata-rata tercatat sebesar 8,8 detik dan gas fee rata-rata sebesar 0.000045 ETH. Efisiensi ini mencerminkan arsitektur kontrak yang ringan dan sesuai standar keamanan.



Gambar 1. Perbandingan Waktu Eksekusi Setiap Aksi Transaksi



Gambar 2. Perbandingan Konsumsi Gas Fee untuk Setiap Aksi

Hasil Audit Keamanan dengan Slither

Audit dilakukan menggunakan Slither Static Analyzer versi terbaru. Analisis ini memeriksa kontrak secara statis untuk mendeteksi kerentanan umum dalam pengembangan smart contract.

Tabel 2. Ringkasan Temuan Audit Keamanan SDS Token

No	Jenis Detektor	Kategori	Temuan	Status
1	Reentrancy	High	Tidak ditemukan	Aman
2	Unchecked Return Value	Medium	Tidak ditemukan	Aman
3	Approval Race Condition	Low	Terdeteksi dan dimitigasi	Aman
4	Unrestricted Minting	Informative	Dibatasi oleh MINTER_ROLE	Aman dengan syarat
5	Compiler Warning	Info	Minor bug pada Solidity ^0.8.20	Diantisipasi

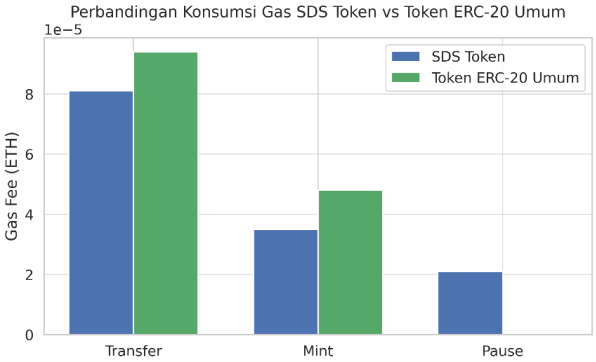
Audit menunjukkan tidak terdapat kerentanan kritis. Temuan minor seperti race condition dan minting dikendalikan melalui mekanisme peran (MINTER_ROLE) yang dirancang sesuai standar keamanan OpenZeppelin.

Perbandingan Efisiensi dengan Token ERC-20 Umum

Sebagai validasi tambahan, dilakukan perbandingan efisiensi konsumsi gas antara SDS Token dan kontrak ERC-20 umum (misal USDT atau kontrak ERC-20 minimalis). Tabel berikut menyajikan data tersebut:

Tabel 3. Perbandingan Gas Fee SDS Token vs Token ERC-20 Umum

Jenis Aksi	SDS Token (Gas Fee ETH)	Token ERC-20 Umum (Gas Fee ETH)	Efisiensi (%)
Transfer	0.000081	0.000094	13,8%
Mint	0.000035	0.000048	27,1%
Pause	0.000021	–	N/A



Gambar 3. Perbandingan Konsumsi Gas SDS Token dan Token ERC-20 Umum

Dari hasil tersebut, SDS Token menunjukkan efisiensi konsumsi gas yang lebih baik. Hal ini memperkuat bahwa kontrak pintar SDS telah dirancang secara optimal dalam hal struktur dan fungsionalitas. Fungsi pause juga menunjukkan konsumsi gas sangat rendah, meskipun belum tersedia dalam kontrak ERC-20 standar untuk dibandingkan secara langsung.

Pembahasan

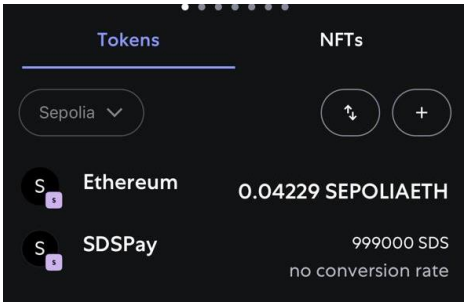
Keberhasilan pengembangan dan implementasi token SDS menunjukkan bahwa desain kontrak pintar berbasis ERC-20 yang mengikuti praktik terbaik arsitektur modular mampu menghasilkan sistem token yang fungsional, fleksibel, dan aman. Penggunaan pustaka standar seperti OpenZeppelin Contracts berkontribusi dalam mengurangi risiko kesalahan struktural serta mendukung penerapan kontrol akses berbasis peran secara efisien [8], [13].

Fitur pause memberikan mekanisme kontrol preventif untuk menghentikan sementara seluruh aktivitas kontrak pada situasi tidak normal, sementara fitur mint dibatasi hanya pada akun yang memiliki MINTER_ROLE, guna mencegah terjadinya inflasi token yang tidak sah. Fitur-fitur tersebut selaras dengan prinsip desain aman yang diadopsi dalam kontrak pintar berbasis Ethereum [14].

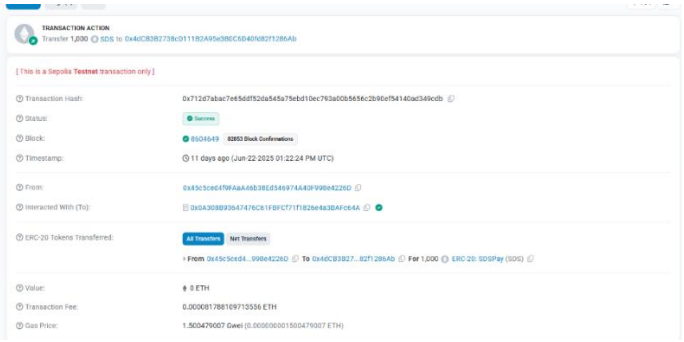
Audit kontrak yang dilakukan menggunakan Slither Static Analyzer mampu mengidentifikasi potensi kerentanan sejak tahap awal pengembangan, termasuk kemungkinan celah reentrancy, penggunaan variabel tak terpakai, dan race condition. Efektivitas analisis statis ini telah dibuktikan dalam berbagai penelitian sebagai salah satu pendekatan paling andal dalam pengujian keamanan kontrak pintar [15].

Dari sisi efisiensi, konsumsi gas selama proses pengujian berada pada kisaran moderat dan dapat diterima untuk konteks implementasi awal di jaringan testnet. Temuan ini sejalan dengan studi sebelumnya yang menekankan pentingnya efisiensi penggunaan ruang penyimpanan dan penghindaran instruksi kompleks dalam struktur kontrak pintar Ethereum [16].

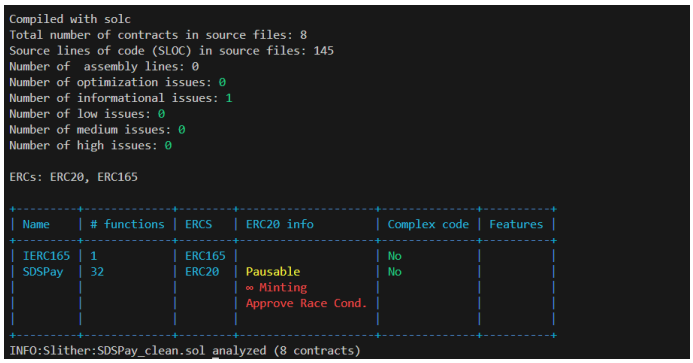
Secara keseluruhan, hasil pengembangan dan pengujian ini menunjukkan bahwa token SDS memiliki potensi untuk dijadikan sebagai model awal dalam pengembangan sistem pembayaran digital berbasis blockchain. Pengembangan lebih lanjut dimungkinkan melalui integrasi fitur tambahan seperti oracle eksternal, mekanisme auto-burn, maupun fungsi staking berbasis waktu sebagai bagian dari peningkatan fungsionalitas sistem di masa mendatang.



Gambar 4. Token SDS terdeteksi otomatis di wallet MetaMask pengguna setelah deployment.



Gambar 5. Detail transaksi transfer SDS pada Etherscan menunjukkan status sukses, waktu eksekusi, dan efisiensi gas.



Gambar 6. Hasil audit statis menggunakan Slither menunjukkan tidak ditemukan kerentanan tinggi pada kontrak SDS.

4. KESIMPULAN DAN SARAN

Kesimpulan

Penelitian ini telah menghasilkan sebuah prototipe token digital bernama SDSPay (SDS) yang dibangun berdasarkan standar ERC-20 dengan integrasi sejumlah fitur lanjutan, antara lain mekanisme penghentian sementara (pause), pencetakan token (mint), serta pengelolaan hak akses berbasis peran. Proses implementasi menunjukkan bahwa kontrak pintar dapat

dijalankan secara optimal pada jaringan uji Ethereum Sepolia Testnet, dengan stabilitas waktu eksekusi transaksi dan konsumsi gas dalam kategori efisien.

Hasil verifikasi dan pengujian fungsional menyatakan bahwa seluruh fungsi utama kontrak telah berjalan sesuai spesifikasi, sedangkan fitur permit telah terintegrasi dalam struktur namun belum diaktifkan secara fungsional. Sementara itu, proses audit menggunakan alat analisis statis Slither tidak menemukan indikasi kerentanan kritis, termasuk reentrancy, race condition, maupun pelanggaran terhadap standar ERC-20.

Temuan ini mendukung validitas arsitektur kontrak yang dirancang, serta memberikan dasar bagi pengembangan lebih lanjut, baik dalam konteks penerapan sistem pembayaran digital internal maupun perluasan ke fungsi lanjutan seperti staking, auto-burn, dan integrasi oracle pada fase implementasi berikutnya.

Saran

Pengembangan lebih lanjut terhadap prototipe kontrak SDS dapat dilakukan untuk meningkatkan skalabilitas, efisiensi, dan fungsionalitas sistem secara menyeluruh. Salah satu pengembangan potensial adalah aktivasi dan pengujian fitur permit (EIP-2612), yang memungkinkan proses otorisasi transaksi dilakukan tanpa biaya gas (gasless approval), sehingga meningkatkan efisiensi dari sisi pengguna.

Selain itu, integrasi dengan antarmuka pengguna berbasis Web3 menjadi penting untuk menguji interaksi end-to-end antara pengguna dan sistem kontrak pintar dalam lingkungan aplikasi nyata. Pengujian pada jaringan uji seperti Sepolia perlu dilanjutkan pada jaringan publik (mainnet), baik di Ethereum maupun alternatif seperti BNB Chain atau Polygon, guna mengevaluasi performa sistem dalam konteks beban jaringan yang lebih kompleks dan bervariasi.

Dari sisi arsitektur token, perlu dipertimbangkan penambahan modul ekonomi seperti mekanisme staking, pembakaran token otomatis, serta integrasi oracle harga sebagai bagian dari dinamika pasar yang lebih kompleks. Untuk menjamin keamanan dalam skala yang lebih besar, disarankan juga dilakukan peninjauan ulang terhadap struktur peran dan hak akses (access control) sesuai dengan kebutuhan operasional aktual, agar sistem tetap adaptif namun tetap aman dalam lingkungan multi-pengguna.

Dengan pengembangan-pengembangan tersebut, token SDS berpotensi untuk diadaptasikan sebagai sistem transaksi digital yang tidak hanya terbatas pada skala internal perusahaan, tetapi juga dapat digunakan dalam ekosistem yang lebih luas dengan tingkat efisiensi dan nilai inovatif yang tinggi.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Faulinda Ely Nastiti, S.Kom., M.Eng., Ph.D selaku pembimbing utama dan Bapak Sopingi, M.Kom selaku pembimbing kedua atas segala bimbingan, arahan, dan evaluasi selama proses penyusunan jurnal ini. Terima kasih juga disampaikan kepada seluruh dosen Politeknik Negeri Malang, serta rekan-rekan mahasiswa yang telah memberikan dukungan, saran, dan semangat selama proses penelitian dan implementasi proyek ini.

Penulis juga mengapresiasi dukungan teknis dari komunitas pengembang Ethereum dan OpenZeppelin, serta layanan testnet publik Sepolia yang memungkinkan proses eksperimen berjalan secara lancar dan bebas biaya.

REFERENSI

Balcerzak, A. P., Nica, E., Rogalska, E., Poliak, M., Klieštik, T., & Sabie, O. M. (2022). Blockchain technology and smart contracts in decentralized governance systems. Administrative Sciences, 12(3). https://doi.org/10.3390/admsci12030096

Cao, X., Zhang, J., Wu, X., & Liu, B. (2022). A survey on security in consensus and smart contracts. Peer-to-Peer Networking and Applications, 15(2), 1008–1028. https://doi.org/10.1007/s12083-021-01268-2

Del Sarto, N., Gai, L., & Ielasi, F. (2024). Financial innovation: The impact of blockchain technologies on financial intermediaries. Journal of Financial Management, Markets, and Institutions, 12(1), 1–21. https://doi.org/10.1142/S2282717X23500056

Fu, X., Wang, H., & Shi, P. (2021). A survey of blockchain consensus algorithms: Mechanism, design and applications. Science China Information Sciences, 64(2), 1–15. https://doi.org/10.1007/s11432-019-2790-1

Islam, M. M., Islam, M. K., Shahjalal, M., Chowdhury, M. Z., & Jang, Y. M. (2023). A low-cost cross-border payment system based on auditable cryptocurrency with consortium blockchain: Joint digital currency. IEEE Transactions on Services Computing, 16(3), 1616–1629. https://doi.org/10.1109/TSC.2022.3207224

Le Quoc, K., et al. (2022). Letter-of-credit chain: Cross-border exchange based on blockchain and smart contracts. International Journal of Advanced Computer Science and Applications, 13(8), 890–898. https://doi.org/10.14569/ijacsa.2022.01308103

Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making smart contracts smarter. Proceedings of the ACM Conference on Computer and Communications Security, 24–28 October, 254–269. https://doi.org/10.1145/2976749.2978309

Nii, R., Sosu, A., Chen, J., Brown-acquaye, W., Owusu, E., & Boahen, E. (2022). A vulnerability detection approach for automated smart contract using enhanced machine learning techniques. Read Full License, A Vulnerability Detection Approach for Automated Smart Contract, 0–10.

Noname. (2021). Central bank digital currencies for cross-border payments: Report to the G20. July.

Oyinkanola, L. O. A., Aremu, O. A., Fajemiroye, J. A., & Makinde, S. O. (2023). On the physical significance and dielectric response of castor oil processed in Nigeria as transformer insulating fluid. International Journal of Research in Advanced Sciences, VIII(2454), 60–66. https://doi.org/10.51584/IJRIAS

Safiullin Kazan, M., & Yelshin, L. (2023). Prospects for using blockchain in the system of international supply chains and cross-border payments. Journal of Management Technology, 23, 360–376.

Signer, C. (2018). Gas cost analysis for Ethereum smart contracts (p. 40). https://doi.org/10.3929/ethz-b-000312914

Transparan, E. Y., Desentralisasi, D. A. N., & Desentralisasi, E. Y. (2025). Penerapan blockchain sebagai solusi untuk sistem administrasi. https://doi.org/10.13140/RG.2.2.31185.03681

Ummah, M. S. (2019). No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析. Sustainability, 11(1), 1–14. http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y

Zetzsche, D. A., Anker-Sørensen, L., Passador, M. L., & Wehrli, A. (2022). DLT-Based enhancement of cross-border payment efficiency – a legal and regulatory perspective. SSRN Electronic Journal, 1015. https://doi.org/10.2139/ssrn.3984523

Zhang, R., Xue, R., & Liu, L. (2019). Security and privacy on blockchain. ACM Computing Surveys, 52(3). https://doi.org/10.1145/3316481