Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Nomor 1, Januari 2026
http://doi.org/10.33395/remik.v10i1.15906
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Analisis Perbandingan Keamanan Smart Contract Ethereum Menggunakan Slither dan Mythril

[1]Iqbal, [2] Hasriadi
[1,2] Program Studi Teknik Informatika, Universitas Lamappapoleonro Soppeng, Indonesia

[1]iqbal@unipol.ac.id [2]hasriadi@unipol.ac.id

*Penulis Korespondensi
Diajukan : 19/01/2026
Diterima : 10/02/2026
Dipublikasi : 13/02/2026

ABSTRACT

The rapid development of blockchain technology, particularly Ethereum, has accelerated the adoption of smart contracts in various decentralized applications. However, the immutable nature of smart contracts once deployed makes security a critical concern. Numerous security incidents have demonstrated that vulnerabilities in smart contracts can lead to significant financial losses and undermine trust in blockchain-based systems. This study aims to analyze and compare the effectiveness of two widely used Ethereum smart contract security analysis tools, Slither and Mythril, in detecting vulnerabilities. An experimental approach with a comparative study design was employed using a dataset of 200 Solidity smart contracts categorized into secure and vulnerable contracts. The evaluation was conducted using precision, recall, F1-score, and analysis time as performance metrics. The results indicate that Slither achieves higher precision and significantly faster analysis time, while Mythril demonstrates higher recall in detecting vulnerable contracts through its symbolic execution approach, albeit with longer analysis duration. These findings reveal a trade-off between efficiency and depth of vulnerability detection. The study concludes that no single tool is superior in all aspects of smart contract security analysis; therefore, combining static analysis and symbolic execution techniques is recommended. Future research may focus on developing hybrid approaches and expanding the dataset to enhance detection coverage and empirical validity.

Keywords: *Blockchain; Ethereum; Mythril; Security Analysis; Slither; Smart Contract; Symbolic Execution.*

ABSTRAK

Perkembangan teknologi blockchain, khususnya Ethereum, telah mendorong pemanfaatan smart contract dalam berbagai aplikasi terdesentralisasi, namun sifatnya yang immutable setelah dideploy menjadikan aspek keamanan sebagai isu yang sangat krusial. Berbagai insiden keamanan menunjukkan bahwa kerentanan pada smart contract dapat menyebabkan kerugian finansial signifikan dan menurunkan kepercayaan terhadap sistem berbasis blockchain. Penelitian ini bertujuan untuk menganalisis dan membandingkan kemampuan dua alat analisis keamanan smart contract Ethereum, yaitu Slither dan Mythril, dalam mendeteksi kerentanan. Metode yang digunakan adalah pendekatan eksperimental dengan desain comparative study terhadap 200 smart contract Solidity yang diklasifikasikan menjadi kontrak aman dan kontrak rentan. Evaluasi dilakukan menggunakan metrik precision, recall, F1-score, serta waktu analisis. Hasil penelitian menunjukkan bahwa Slither memiliki tingkat precision yang lebih tinggi dan waktu analisis yang lebih cepat, sedangkan Mythril memiliki recall yang lebih tinggi dalam mendeteksi kontrak rentan melalui pendekatan symbolic execution, meskipun memerlukan waktu analisis lebih lama. Temuan ini menunjukkan adanya trade-off antara efisiensi dan kedalaman deteksi. Kesimpulan penelitian menegaskan bahwa tidak terdapat satu alat yang sepenuhnya unggul dalam seluruh aspek analisis keamanan, sehingga kombinasi analisis statis dan symbolic execution direkomendasikan. Penelitian

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

selanjutnya dapat mengembangkan pendekatan hibrida dan memperluas dataset untuk meningkatkan validitas dan cakupan deteksi kerentanan.

Kata Kunci: Analisis Keamanan, Blockchain, Ethereum, Mythril, Smart Contract, Slither.

I. PENDAHULUAN

Perkembangan teknologi blockchain telah mengubah paradigma sistem transaksi digital terdesentralisasi sejak diperkenalkan melalui Bitcoin oleh Nakamoto (2008). Evolusi tersebut semakin signifikan dengan hadirnya Ethereum yang memungkinkan implementasi *smart contract*, yaitu program yang berjalan secara otomatis di atas blockchain tanpa perantara (Buterin, 2014; Wood, 2014). Smart contract kini menjadi fondasi berbagai aplikasi modern seperti decentralized finance (DeFi), tokenisasi aset, NFT, serta sistem perdagangan kripto global (Li et al., 2021; Zhang et al., 2021).

Meskipun menawarkan transparansi dan otomatisasi, karakter *immutable* smart contract menjadikan aspek keamanan sebagai tantangan krusial. Kesalahan logika atau celah keamanan tidak dapat diperbaiki setelah kontrak dideploy, sehingga berpotensi menimbulkan kerugian finansial besar (Liu et al., 2021). Studi empiris menunjukkan bahwa eksploitasi terhadap kontrak pintar masih sering terjadi akibat kelemahan desain maupun kesalahan implementasi (Garcia et al., 2023; Khan et al., 2025). Kerentanan seperti reentrancy, integer overflow, dan improper access control tetap menjadi ancaman dominan dalam ekosistem blockchain (Zhou et al., 2022).

Untuk mengatasi permasalahan tersebut, berbagai pendekatan analisis keamanan dikembangkan, termasuk *static analysis*, *symbolic execution*, serta pendekatan hibrida (Torres et al., 2024). Static analysis memungkinkan deteksi cepat tanpa mengeksekusi kontrak, sedangkan symbolic execution mengeksplorasi jalur eksekusi untuk mengidentifikasi kondisi kerentanan yang lebih kompleks (Wang et al., 2022; Khan et al., 2025). Dua alat yang banyak digunakan dalam praktik dan penelitian adalah Slither dan Mythril. Slither berbasis static analysis dan dikenal efisien dalam mendeteksi pola kerentanan umum (Feist et al., 2019), sementara Mythril menggunakan symbolic execution untuk eksplorasi jalur eksekusi yang lebih mendalam (Consensys, 2023).

Meskipun sejumlah penelitian telah mengevaluasi alat-alat tersebut, studi komparatif yang menggunakan dataset terklasifikasi secara sistematis masih relatif terbatas (Kumar et al., 2024). Benchmarking terbaru menunjukkan adanya variasi signifikan dalam precision dan recall antar alat analisis (Garcia et al., 2023), sehingga diperlukan evaluasi empiris yang terkontrol untuk memahami performa relatif masing-masing pendekatan.

Dalam konteks Indonesia, peningkatan adopsi blockchain belum sepenuhnya diimbangi dengan kesiapan keamanan teknis (Rahman & Wibowo, 2022). Dinamika pasar kripto nasional yang semakin kompleks (Hasriadi & Iqbal, 2026) menegaskan pentingnya evaluasi keamanan smart contract sebagai fondasi kepercayaan sistem.

Berdasarkan gap tersebut, penelitian ini bertujuan membandingkan kemampuan Slither dan Mythril dalam mendeteksi kerentanan smart contract Ethereum menggunakan dataset yang diklasifikasikan sebagai aman dan rentan. Evaluasi dilakukan menggunakan metrik precision, recall, F1-score, serta efisiensi waktu analisis untuk menghasilkan rekomendasi praktis bagi pengembang dan peneliti keamanan blockchain.

II. TINJAUAN PUSTAKA

Smart Contract dan Ekosistem Ethereum

Smart contract merupakan program terdesentralisasi yang dijalankan oleh Ethereum Virtual Machine (EVM) dan memungkinkan otomatisasi logika bisnis berbasis blockchain (Wood, 2014). Perkembangan aplikasi DeFi dan tokenisasi aset telah meningkatkan kompleksitas kontrak, sehingga risiko kesalahan logika dan eksploitasi turut meningkat (Li et al., 2021).

Kerentanan Smart Contract

Penelitian mutakhir menunjukkan bahwa kerentanan smart contract masih menjadi isu dominan dalam keamanan blockchain (Liu et al., 2021). Kerentanan umum meliputi reentrancy,

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

integer overflow, denial of service, dan improper authorization (Garcia et al., 2023). Untuk standarisasi, Smart Contract Weakness Classification (SWC) Registry dikembangkan sebagai referensi kategorisasi kerentanan (SWC Registry, 2023). Penggunaan klasifikasi ini meningkatkan konsistensi evaluasi penelitian dan audit keamanan.

Static Analysis

Static analysis menganalisis kode sumber tanpa eksekusi program. Pendekatan ini efektif dalam mendeteksi pola kerentanan sintaksis dan struktural (Wang et al., 2022). Integrasi static analyzer dalam pipeline CI/CD terbukti meningkatkan keamanan pengembangan perangkat lunak blockchain (Zhang et al., 2021). Namun, pendekatan ini memiliki keterbatasan dalam mendeteksi kerentanan berbasis kondisi runtime kompleks (Torres et al., 2024).

Symbolic Execution

Symbolic execution menggantikan input konkret dengan variabel simbolik untuk mengeksplorasi jalur eksekusi (Zhou et al., 2022). Teknik ini mampu mengidentifikasi kerentanan tersembunyi dalam kombinasi logika kompleks (Khan et al., 2025). Tantangan utama adalah fenomena *path explosion* yang meningkatkan waktu komputasi.

Slither

Slither merupakan alat analisis statis yang mengubah kode Solidity menjadi intermediate representation sehingga mempermudah analisis alur kontrol dan data flow (Feist et al., 2019). Slither dapat mendeteksi kerentanan umum seperti reentrancy, unprotected functions, dan unsafe math operations. Kecepatan serta kemudahan integrasinya menjadikan Slither alat yang populer di kalangan pengembang Ethereum.

Mythril

Mythril adalah alat analisis keamanan yang menggunakan symbolic execution untuk menguji kemungkinan jalur eksekusi kontrak (Consensys, 2023). Dengan analisis yang lebih mendalam, Mythril mampu mengidentifikasi kerentanan tingkat lanjut, namun di sisi lain memerlukan waktu eksekusi yang lebih besar dibandingkan alat statis seperti Slither.

Evaluasi Alat Analisis

Benchmarking terhadap berbagai alat analisis menunjukkan tidak ada satu alat pun yang unggul dalam semua metrik evaluasi (Garcia et al., 2023; Kumar et al., 2024). Oleh karena itu, studi komparatif berbasis dataset terklasifikasi diperlukan untuk memperoleh gambaran objektif mengenai performa masing-masing pendekatan. Penelitian terbaru juga menyoroti pentingnya reliability analysis dalam evaluasi alat keamanan blockchain. Park et al. (2023) menunjukkan bahwa variasi hasil deteksi antar alat dapat berdampak pada tingkat keandalan sistem secara keseluruhan, terutama dalam konteks aplikasi finansial terdesentralisasi.

Penelitian Terdahulu

Sejumlah penelitian telah menyoroti efektivitas dan keterbatasan alat-alat analisis keamanan smart contract. Torres, Steichen, dan State (2024) melakukan tinjauan sistematis terhadap teknik deteksi kerentanan pada smart contract dan menemukan bahwa pendekatan analisis seperti static analysis, symbolic execution, dan kombinasi keduanya memiliki dampak yang berbeda terhadap cakupan deteksi kerentanan. Studi ini menekankan kebutuhan evaluasi komparatif yang lebih terstruktur agar dapat memahami keunggulan dan keterbatasan masing-masing metode secara objektif.

Kumar, Sharma, dan Singh (2024) mengevaluasi beberapa alat analisis keamanan smart contract termasuk Slither dan Mythril dalam konteks Ethereum. Hasil penelitian menunjukkan bahwa tidak ada satu pun alat yang unggul dalam semua metrik evaluasi, dan masing-masing alat memiliki

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

karakteristik deteksi spesifik yang bergantung pada pendekatan analisisnya. Temuan ini menjadi dasar kuat untuk melakukan studi komparatif yang fokus pada evaluasi kinerja alat tertentu dengan dataset yang terverifikasi.

Di konteks lokal, penelitian oleh Rahman dan Wibowo (2022) mengkaji tingkat kesadaran dan praktik keamanan smart contract di kalangan pengembang perangkat lunak di Indonesia. Studi ini mengungkap bahwa banyak pengembang lokal belum secara konsisten menerapkan analisis keamanan otomatis saat mengembangkan smart contract, sehingga potensi risiko kerentanan tetap tinggi. Temuan tersebut mendukung perlunya penelitian empiris yang menguji efektivitas alat-alat analisis keamanan dalam praktik nyata.

III. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dengan desain *comparative study* untuk mengevaluasi performa Slither dan Mythril dalam mendeteksi kerentanan smart contract. Pendekatan eksperimental banyak digunakan dalam penelitian keamanan perangkat lunak untuk memastikan validitas dan reprodusibilitas hasil (Garcia et al., 2023).

Desain Penelitian

Desain penelitian yang digunakan adalah *comparative study*, yang bertujuan untuk mengevaluasi perbedaan kemampuan deteksi kerentanan dan efisiensi analisis antara Slither dan Mythril. Studi perbandingan banyak digunakan dalam penelitian keamanan smart contract untuk mengidentifikasi keunggulan dan keterbatasan masing-masing alat analisis, mengingat tidak ada satu alat pun yang mampu mendeteksi seluruh jenis kerentanan secara sempurna (Kumar et al., 2024). Kedua alat diuji menggunakan dataset smart contract yang sama untuk memastikan keadilan (*fair comparison*) dan konsistensi hasil pengujian.

Dataset Penelitian

Dataset yang digunakan dalam penelitian ini terdiri dari kumpulan smart contract Ethereum berformat Solidity (.sol) yang diklasifikasikan ke dalam dua kategori, yaitu smart contract aman dan smart contract rentan. Smart contract aman diperoleh dari pustaka OpenZeppelin yang telah digunakan secara luas dan melalui proses audit keamanan, sehingga sering dijadikan *baseline* kontrak aman dalam penelitian akademik (Atzei et al., 2017).

Sementara itu, smart contract rentan dikumpulkan dari sumber yang mendokumentasikan kerentanan secara eksplisit, seperti Smart Contract Weakness Classification (SWC) Registry dan repositori kontrak rentan yang digunakan dalam penelitian sebelumnya (SWC Registry, 2023; Torres et al., 2024). Penggunaan dataset yang terklasifikasi secara jelas bertujuan untuk mengurangi ambiguitas dalam evaluasi hasil deteksi dan meningkatkan validitas penelitian.

Dataset Penelitian

Dataset terdiri dari smart contract Ethereum (.sol) yang diklasifikasikan sebagai aman dan rentan. Kontrak aman diperoleh dari pustaka OpenZeppelin yang telah diaudit secara luas. Kontrak rentan dikumpulkan dari SWC Registry dan dataset penelitian sebelumnya (Torres et al., 2024). Pendekatan ini mengikuti praktik penelitian empiris modern dalam evaluasi analyzer (Kumar et al., 2024).

Alat dan Lingkungan Pengujian

Pengujian dilakukan menggunakan dua alat analisis keamanan, yaitu Slither sebagai alat analisis statis dan Mythril sebagai alat analisis berbasis symbolic execution. Slither dipilih karena kemampuannya dalam mendeteksi kerentanan umum secara cepat melalui analisis struktur kode dan alur data, sedangkan Mythril dipilih karena kemampuannya mengeksplorasi jalur eksekusi kompleks menggunakan symbolic execution (Feist et al., 2019; Consensys, 2023).

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

Kedua alat dijalankan pada lingkungan pengujian yang sama untuk menjaga konsistensi hasil. Lingkungan pengujian meliputi sistem operasi berbasis Linux, versi Solidity compiler yang kompatibel dengan smart contract yang diuji, serta dependensi pendukung yang direkomendasikan oleh masing-masing alat analisis.

Prosedur Pengujian

Prosedur pengujian dilakukan melalui beberapa tahap. Pertama, seluruh smart contract dalam dataset dikompilasi untuk memastikan kompatibilitas dengan versi Solidity yang digunakan. Kedua, setiap smart contract dianalisis menggunakan Slither dengan menjalankan konfigurasi analisis standar untuk mendeteksi kerentanan keamanan. Ketiga, smart contract yang sama dianalisis menggunakan Mythril untuk mengeksplorasi jalur eksekusi dan mendeteksi potensi kerentanan berbasis kondisi runtime. Hasil analisis dari masing-masing alat kemudian dikumpulkan dan didokumentasikan, termasuk jenis kerentanan yang terdeteksi, jumlah temuan, serta waktu analisis, untuk selanjutnya dianalisis secara komparatif.

Teknik Analisis Data

Hasil pengujian dianalisis secara deskriptif dan komparatif. Analisis deskriptif digunakan untuk mengidentifikasi jenis dan distribusi kerentanan yang terdeteksi oleh masing-masing alat berdasarkan klasifikasi SWC. Analisis komparatif dilakukan untuk membandingkan kinerja Slither dan Mythril menggunakan beberapa metrik evaluasi yang umum digunakan dalam penelitian keamanan perangkat lunak (Durieux et al., 2018), yaitu:

True Positive (TP): jumlah smart contract rentan yang berhasil terdeteksi

False Positive (FP): jumlah smart contract aman yang terdeteksi sebagai rentan

False Negative (FN): jumlah smart contract rentan yang tidak terdeteksi

Berdasarkan nilai tersebut, digunakan metrik evaluasi sebagai berikut:

Precision

Mengukur tingkat ketepatan alat dalam mendeteksi kerentanan

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall

Mengukur kemampuan alat mendeteksi seluruh kontrak rentan

$$Recall = \frac{TP}{TP + FP} \quad (2)$$

F1-Score

Menyeimbangkan precision dan recall

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Metrik ini umum digunakan dalam evaluasi deteksi kerentanan perangkat lunak (Khan et al., 2025). Selain itu, waktu analisis rata-rata juga dianalisis untuk mengevaluasi efisiensi masing-masing alat.

Validitas Penelitian

Untuk menjaga validitas penelitian, digunakan dataset yang terverifikasi dan prosedur pengujian yang konsisten. Seluruh pengujian dilakukan pada dataset yang sama dengan konfigurasi

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

lingkungan yang identik untuk kedua alat analisis. Selain itu, hasil pengujian dievaluasi berdasarkan klasifikasi kerentanan SWC yang telah diakui secara luas dalam penelitian dan praktik audit keamanan smart contract (SWC Registry, 2023). Pendekatan ini diharapkan dapat meningkatkan objektivitas, reprodusibilitas, dan keandalan hasil penelitian. Pendekatan benchmarking berbasis dataset terklasifikasi telah direkomendasikan dalam berbagai studi keamanan perangkat lunak modern untuk meningkatkan reprodusibilitas dan objektivitas hasil (Garcia et al., 2023; Almeida et al., 2022). Oleh karena itu, penelitian ini mengadopsi prosedur pengujian yang konsisten dan terdokumentasi untuk memastikan validitas internal dan eksternal.

IV. HASIL DAN PEMBAHASAN

Implementasi Sistem

Pada tahap implementasi, penelitian ini menggunakan dua alat analisis keamanan smart contract Ethereum, yaitu Slither dan Mythril. Slither dioperasikan sebagai alat analisis statis yang memeriksa struktur kode Solidity untuk mendeteksi pola kerentanan umum tanpa mengeksekusi kontrak. Sementara itu, Mythril diimplementasikan dengan pendekatan symbolic execution yang memungkinkan eksplorasi berbagai jalur eksekusi kontrak berdasarkan kondisi logis yang mungkin terjadi saat runtime. Kedua alat dijalankan pada lingkungan pengujian yang sama untuk memastikan konsistensi hasil analisis. Implementasi ini memungkinkan perbandingan langsung terhadap karakteristik deteksi kerentanan yang dihasilkan oleh masing-masing alat.

Dataset dan Skenario Pengujian

Dataset yang digunakan terdiri dari smart contract Ethereum berformat Solidity (.sol) yang telah diklasifikasikan ke dalam dua kategori, yaitu kontrak aman dan kontrak rentan. Kontrak aman diperoleh dari repositori proyek open-source yang telah diaudit, sedangkan kontrak rentan diambil dari dataset publik yang secara eksplisit mengandung kerentanan keamanan. Seluruh smart contract dalam dataset telah melalui proses validasi untuk memastikan kompatibilitas dengan versi Solidity yang digunakan pada saat pengujian. Pengujian dilakukan secara terpisah menggunakan Slither dan Mythril pada dataset yang sama untuk memperoleh hasil yang dapat dibandingkan secara objektif.

Hasil Deteksi Kerentanan

Tabel 1. Hasil Pengujian Keamanan Smart Contract Menggunakan Slither dan Mythril

Kategori Kontrak	Tools	Jumlah Kontrak	Terdeteksi Rentan	Tidak Terdeteksi	False Positive	Waktu Analisis (detik)
Aman	Slither	100	8	92	8	4,5
Aman	Mythril	100	15	85	15	39,1
Rentan	Slither	100	82	18	–	5,2
Rentan	Mythril	100	90	10	–	46,8

Tabel 1 menunjukkan hasil pengujian keamanan terhadap 200 smart contract Ethereum yang terdiri dari 100 kontrak aman dan 100 kontrak rentan. Pengujian dilakukan menggunakan Slither dan Mythril pada dataset yang sama untuk memastikan konsistensi perbandingan. Pada kategori kontrak aman, Slither mendeteksi 8 kontrak sebagai rentan, sedangkan Mythril mendeteksi 15 kontrak sebagai rentan. Perbedaan ini menunjukkan bahwa Mythril memiliki tingkat sensitivitas yang lebih tinggi terhadap potensi isu keamanan, namun juga menghasilkan jumlah false positive yang lebih besar dibandingkan Slither.

Pada kategori kontrak rentan, Mythril berhasil mendeteksi 90 kontrak rentan, sementara Slither mendeteksi 82 kontrak rentan. Hasil ini mengindikasikan bahwa pendekatan symbolic execution yang digunakan Mythril lebih efektif dalam mendeteksi kerentanan yang bergantung pada jalur eksekusi tertentu. Namun demikian, keunggulan ini diimbangi dengan waktu analisis yang lebih

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Nomor 1, Januari 2026
http://doi.org/10.33395/remik.v10i1.15906
e-ISSN : 2541-1330
p-ISSN : 2541-1332

lama. Rata-rata waktu analisis Mythril hampir sepuluh kali lebih besar dibandingkan Slither, yang mencerminkan perbedaan kompleksitas pendekatan analisis yang digunakan.

Tabel 2. Evaluasi Kinerja Slither dan Mythril Menggunakan Metrik Klasifikasi

Tools	Precision	Recall	F1-Score
Slither	0,91	0,82	0,86
Mythril	0,86	0,90	0,88

Berdasarkan Tabel 2, Slither menunjukkan nilai precision yang lebih tinggi, yang berarti alat ini lebih akurat dalam mengklasifikasikan kontrak aman dan menghasilkan false positive yang relatif rendah. Sebaliknya, Mythril memiliki nilai recall yang lebih tinggi, menunjukkan kemampuannya dalam mendeteksi lebih banyak kontrak rentan. Nilai F1-score pada kedua alat menunjukkan bahwa performa keduanya relatif seimbang, namun dengan fokus dan keunggulan yang berbeda.

Secara keseluruhan, hasil evaluasi menegaskan adanya trade-off antara efisiensi waktu dan kedalaman analisis. Slither lebih sesuai digunakan untuk analisis awal atau pengujian berskala besar yang membutuhkan kecepatan, sedangkan Mythril lebih tepat digunakan untuk analisis mendalam terhadap smart contract dengan kompleksitas logika tinggi sebelum proses deployment. Temuan ini memperkuat pentingnya pemilihan alat analisis keamanan smart contract yang disesuaikan dengan kebutuhan dan konteks penggunaan.

V. KESIMPULAN

Penelitian ini telah melakukan analisis perbandingan keamanan smart contract Ethereum menggunakan dua alat analisis yang berbeda pendekatan, yaitu Slither dan Mythril. Berdasarkan hasil pengujian terhadap 200 smart contract yang terdiri dari kontrak aman dan kontrak rentan, diperoleh temuan bahwa kedua alat memiliki karakteristik dan kinerja yang berbeda dalam mendeteksi kerentanan keamanan. Slither menunjukkan keunggulan dalam efisiensi waktu analisis dan tingkat ketepatan klasifikasi yang lebih baik pada smart contract aman, sehingga menghasilkan jumlah false positive yang relatif rendah. Sebaliknya, Mythril memiliki kemampuan deteksi yang lebih tinggi terhadap smart contract rentan, khususnya kerentanan yang bergantung pada kondisi logika dan jalur eksekusi tertentu, meskipun membutuhkan waktu analisis yang lebih lama. Hasil evaluasi menggunakan metrik precision, recall, dan F1-score menegaskan adanya trade-off antara kecepatan analisis dan kedalaman deteksi kerentanan. Temuan ini menunjukkan bahwa tidak terdapat satu alat yang sepenuhnya unggul dalam seluruh aspek analisis keamanan smart contract Ethereum. Oleh karena itu, pemilihan alat analisis keamanan sebaiknya disesuaikan dengan kebutuhan dan konteks penggunaan dalam proses pengembangan aplikasi berbasis blockchain. Berdasarkan hasil penelitian, disarankan agar pengembang dan peneliti di bidang Informatika mempertimbangkan penggunaan kombinasi alat analisis statis dan symbolic execution dalam proses pengujian keamanan smart contract untuk memperoleh cakupan deteksi kerentanan yang lebih komprehensif. Penelitian selanjutnya dapat mengembangkan pendekatan hibrida dengan mengintegrasikan Slither dan Mythril ke dalam pipeline pengembangan perangkat lunak, seperti continuous integration pada sistem blockchain. Selain itu, penelitian di masa depan dapat memperluas dataset dengan jumlah dan variasi smart contract yang lebih besar, serta mempertimbangkan penggunaan alat analisis dinamis untuk melengkapi hasil pengujian statis. Pengembangan kerangka evaluasi yang lebih terstandar dan otomatis juga berpotensi memberikan kontribusi terhadap peningkatan kualitas keamanan smart contract dan inovasi teknologi blockchain di bidang Informatika.

VI. REFERENSI

Almeida, J., Cruz, P., & Henriques, M. (2022). Detecting vulnerabilities in Ethereum smart contracts: An empirical evaluation. *Information and Software Technology, 146*, 106–118.

Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts. In *Principles of Security and Trust* (pp. 164–186)

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform*. Ethereum White Paper.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

Consensys. (2023). *Mythril classic: Security analysis tool for EVM bytecode*. Retrieved from

Durieux, T., Ferreira, J. F., Abreu, R., & Cruz, P. (2018). Empirical review of automated analysis tools on smart contracts. In *Proceedings of the 40th International Conference on Software Engineering* (pp. 1–12). ACM.

Feist, J., Grieco, G., & Groce, A. (2019). Slither: A static analysis framework for smart contracts. In *Proceedings of the IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain* (pp. 8–15). IEEE.

Garcia, D., Torres, C., & State, R. (2023). Benchmarking smart contract security analyzers. *IEEE Access, 11*, 55321–55338.

Hasriadi, & Iqbal. (2026). Prediksi harga Bitcoin multivariat OHLC berbasis hybrid LSTM–GRU dan sentimen berita. *Jurnal Manajemen dan Penelitian, 10*(1), 1–15.

Khan, M., Lee, S., & Park, J. (2025). Evaluating symbolic execution tools for Ethereum smart contracts. *Computers & Security, 135*, 103–120.

Kumar, R., Sharma, P., & Singh, A. (2024). Comparative evaluation of smart contract analysis tools. In *Proceedings of the IEEE International Conference on Blockchain* (pp. 112–119). IEEE.

Liu, Z., Zhang, H., & Chen, Y. (2021). Smart contract security: A systematic literature review. *Journal of Systems Architecture, 117*, 102–120.

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*.

Park, J., Kim, H., & Lee, S. (2023). Security evaluation of blockchain smart contracts using hybrid analysis techniques. *IEEE Transactions on Reliability, 72*(3), 1187–1199.

Rahman, F., & Wibowo, A. (2022). Smart contract security awareness in Indonesia: An empirical study. *Jurnal Teknologi Informasi dan Ilmu Komputer, 9*(2), 101–110.

SWC Registry. (2023). *Smart contract weakness classification registry*. Retrieved from https://swcregistry.io

Torres, C., Steichen, M., & State, R. (2024). Smart contract security analysis: A systematic literature review. *IEEE Access, 12*, 1–15.

Wang, S., Liu, H., & Zhang, X. (2022). A comparative study of smart contract analysis tools. In *Proceedings of the IEEE International Conference on Blockchain* (pp. 112–119). IEEE.

Wood, G. (2014). *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum Yellow Paper.

Zhang, Y., Lou, J., Chen, Q., & Li, Z. (2021). Smart contract vulnerability detection using graph neural networks. *IEEE Transactions on Dependable and Secure Computing, 18*(6), 3245–3258.

Zhou, Y., Kumar, D., & Wang, Y. (2022). Hybrid static and dynamic analysis for smart contract security. In *Lecture Notes in Computer Science* (Vol. 133, pp. 133–148).

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.