INTISARI

Penelitian ini bertujuan untuk merancang dan mengevaluasi alur kerja berbasis konsep *secure coding*. Alur ini dibangun di atas Git dan secara otomatis mengintegrasikan pengujian keamanan statis (SAST) menggunakan Semgrep dan pengujian dinamis (DAST) menggunakan OWASP ZAP. Tujuannya adalah mencegah integrasi kode rentan ke dalam cabang utama pengembangan. Evaluasi dilakukan dengan mengukur efisiensi waktu proses serta menguji beberapa skenario penggabungan kode. Alur yang dibangun juga disesuaikan dengan standar ISO/IEC 27001:2022 terkait. Hasil menunjukkan bahwa alur otomatis berjalan efisien dan mampu mencegah integrasi kode yang mengandung kerentanan secara otomatis.

Sebagai pelengkap, dilakukan pula evaluasi efektivitas Semgrep dan ZAP terhadap dataset *OWASP Benchmark* untuk menilai kemampuan deteksi masing-masing. Semgrep menunjukkan cakupan deteksi tinggi (recall 88,20%) namun menghasilkan cukup banyak *false positive*. Sebaliknya, ZAP memiliki tingkat ketepatan deteksi sangat tinggi (precision 97,77%) namun dengan cakupan terbatas (recall 27,84%). Temuan ini menunjukkan bahwa kedua alat memiliki kelebihan dan kekurangan masing-masing, serta lebih efektif jika digunakan secara komplementer.

Penelitian ini memberikan kontribusi berupa kerangka kerja otomatis *secure - coding* yang dapat diintegrasikan dalam alur pengembangan perangkat lunak modern, serta analisis perbandingan terhadap dua pendekatan utama pengujian keamanan.

Kata kunci: Secure Coding, SAST, DAST, Semgrep, OWASP ZAP, ISO/IEC 27001:2022, Git

ABSTRACT

This study aims to design and evaluate a Git-based workflow grounded in secure coding principles. The proposed workflow integrates automated static (SAST) and dynamic (DAST) security testing using Semgrep and OWASP ZAP, respectively. It is designed to enhance software security by preventing the integration of vulnerable code into the main development branch. The evaluation involved measuring workflow execution time and testing various merge scenarios. The workflow developed is also aligned with the relevant ISO/IEC 27001:2022 standards. Results show that the automated workflow runs efficiently and can effectively block the integration of insecure code.

Additionally, the effectiveness of Semgrep and ZAP was assessed separately using the OWASP Benchmark dataset to evaluate their detection capabilities. Semgrep achieved high detection coverage (recall 88.20%) but produced a significant number of false positives. In contrast, ZAP demonstrated excellent detection accuracy (precision 97.77%) but had limited coverage (recall 27.84%). These findings indicate that each tool has its own strengths and limitations, and they are more effective when used complementarily.

This research contributes an automated secure coding framework that can be integrated into modern development workflows, along with a comparative analysis of two major security testing approaches.

Keywords: Secure Coding, SAST, DAST, Semgrep, OWASP ZAP, ISO/IEC 27001:2022, Git