ABSTRACT

Software security is a vital aspect of modern application development, especially amidst the increasing cyber threats that cause global losses of $10.5 trillion per year with 15% annual growth. To address this challenge, this research developed a web application designed to detect vulnerabilities in source code by leveraging the integration between Static Application Security Testing (SAST) via the Bearer CLI and the analysis capabilities of Large Language Models (LLM) such as GPT-4. The application allows users to upload code, perform vulnerability analysis, and obtain remediation recommendations relevant to the vulnerabilities found. The research methodology includes the stages of requirements analysis, system design, implementation, and testing using Black Box Testing and accuracy evaluation with OWASP Benchmark and Skf labs datasets. The results showed that the application was successfully developed with a functional testing success rate of 100%. In terms of accuracy, Bearer CLI showed variations between 45.04% to 90.69%, while the LLM GPT-4 model achieved 85% accuracy, 88.88% precision, and 94.11% recall. However, the low specificity (33.33%) indicates the need to reduce false positives. This research proves that the integration of SAST and LLM is effective in detecting vulnerabilities and provides recommendations for improvement. Despite the good results achieved, this research highlights the need for further improvements to reduce false positives in LLM and increase the accuracy of SAST to strengthen overall software quality and security.

Keywords: *software security, SAST, Bearer CLI, Large Language Model, security testing, vulnerability detection.*