IMPLEMENTASI DEVSECOPS DENGAN METODE STATIC APPLICATION SECURITY TESTING (SAST) MENGGUNAKAN SNYK PADA APLIKASI BERBASIS CONTAINER

M. Zhafran Rayhan [1] **, Muhammad Arif Fadhly Ridha** [2]

[1,2] *Program Studi Teknik Informatika, Politeknik Caltex Riau, Pekanbaru, 28265*
E-mail: [1)] zhafran19ti@mahasiswa.pcr.ac.id [2)] fadhly@pcr.ac.id

Abstract

This research proposes the use of DevSecOps with the Static Application Security Testing (SAST) approach using the Snyk platform to enhance efficiency and security in the software development process. The SAST methodology enables testing of potential cybersecurity exploits during the system's building and maintenance phases. By employing Snyk, a security scanning platform that can integrate with Integrated Development Environments (IDEs) and support container or cloud-based applications, developers can automatically and comprehensively scan their code, Quantitative testing was conducted by scanning 10 websites of Politeknik Caltex Riau, revealing a total of 1089 vulnerabilities, with the majority falling into the "Low" category. These findings indicate that low-level vulnerabilities dominate the tested systems. Meanwhile, qualitative testing was performed through interviews with programmers as the respondents. During the interviews, programmers stated that the use of Snyk SAST in the development process allowed them to detect security gaps before releasing to the public. They also found Snyk's recommendations and suggestions valuable for making improvements.

Keywords: *DevSecOps, Static Application Security Testing (SAST), Snyk, IDE, security vulnerabilities.*

Abstrak

Penelitian ini mengusulkan penggunaan DevSecOps dengan pendekatan Static Application Security Testing (SAST) menggunakan platform Snyk untuk meningkatkan efisiensi dan keamanan dalam proses pengembangan perangkat lunak. Metode SAST memungkinkan pengujian potensi eksploitasi keamanan siber dalam fase pembuatan dan pemeliharaan sistem. Dengan Snyk, platform pemindaian keamanan yang dapat terintegrasi dengan Integrated Development Environment (IDE) serta mendukung aplikasi berbasis kontainer atau awan, pengembang dapat melakukan pemindaian kode program secara otomatis dan menyeluruh, Hasil pengujian kuantitatif dilakukan dengan memindai 10 (sepuluh) situs web Politeknik Caltex Riau dan ditemukan total 1089 kerentanan, dengan sebagian besar termasuk dalam kategori "Rendah". Temuan ini menunjukkan bahwa kerentanan tingkat rendah mendominasi sistem yang diuji. Sementara itu, hasil pengujian kualitatif dilakukan melalui wawancara dengan pemrogram sebagai narasumber. Dalam wawancara tersebut, pemrogram menyatakan bahwa penggunaan Snyk SAST dalam pengembangan sistem memungkinkan mereka untuk mendeteksi celah keamanan sebelum dirilis ke publik. Mereka juga merasa bahwa Snyk memberikan rekomendasi dan saran yang berguna untuk perbaikan. Secara keseluruhan, penggunaan Snyk dalam alur kerja DevSecOps terbukti efektif dalam mengidentifikasi kerentanan keamanan dalam pengembangan perangkat lunak.

Kata kunci: *DevSecOps, Static Application Security Testing (SAST), Snyk, IDE, kerentanan keamanan.*

PENDAHULUAN

Cloud Computing adalah sistem terukur dari kumpulan sumber daya bersama dengan bantuan virtualisasi (Kumar & Rathore, 2018), Teknologi *cloud* dan virtualisasi menjadi semakin kompleks dan sulit untuk dikelola, tetapi salah satu kelemahan perangkat lunak yang dikembangkan adalah tim pengembangan (*Developer*) dan operasi (*Operations*) yang berbeda. dengan proses desain dan pengujian yang terpisah. Cybersecurity telah menjadi bagian penting dan penting dari masyarakat kita. Perangkat lunak sering kali dikembangkan dengan sedikit atau tanpa komponen dan pedoman keamanan, yang memengaruhi kualitas dan keamanan perangkat lunak. (Zunnurhain & Duclervil, 2019),Situasi yang dapat disimpulkan di sini adalah tim pengembangan membuat aplikasi baru dan memodifikasi aplikasi yang ada agar lebih baik dan lebih efektif. Sementara itu, tim operasi memastikan semuanya bekerja dengan baik.

DevOps adalah gabungan dari *Development* dan *Operations*. *DevOps* adalah sebuah prinsip *developer* untuk mengkoordinasikan antara tim *development* dengan tim *operations* dengan efektif dan efisien (Dharma, 2020). *DevOps* adalah budaya yang mempromosikan kerja tim dan mendorong perusahaan untuk menghadirkan fungsionalitas produk perangkat lunak melalui proses otomatis (Bijwe & Shankar, 2022). Tujuan *DevOps* adalah untuk meningkatkan kolaborasi antara tim pengembangan dan operasi dalam pengembangan dan pengiriman perangkat lunak yang terus dirilis. *DevOps* juga memfasilitasi kolaborasi yang lebih luas secara budaya dengan menghadirkan alat dan praktik desain baru ke tim pengembangan. Integrasi Berkelanjutan,

Menurut (Jeganathan, 2019) *DevSecOps* (*Development Security Operations*) relatif baru di bidang informasi keamanan. Istilah *DevSecOps* digunakan untuk memfokuskan tim teknis dan pengembang perusahaan pada keamanan siber untuk meningkatkan keamanan pengembangan aplikasi. Penambahan area fokus keamanan siber akan memungkinkan *DevSecOps* menyediakan layanan keamanan siber yang berfokus pada bisnis untuk menguji potensi eksploitasi keamanan siber

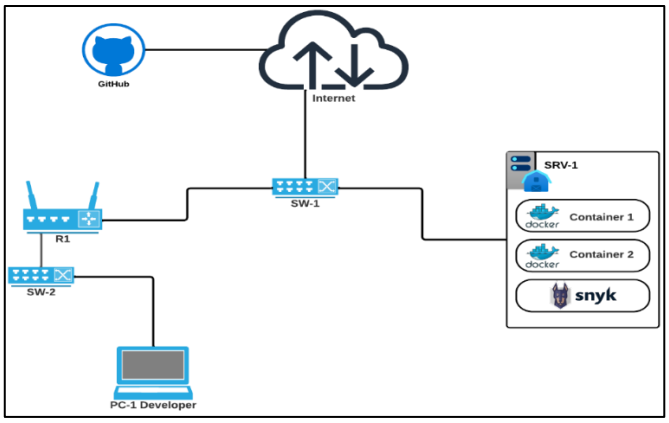
Dalam lingkungan *Cloud Computing* dan *Big data*, masalah keamanan data telah menjadi fokus perhatian(Wang et al., 2021). Keamanan data diperlukan saat merancang dan membangun perangkat lunak dengan semua alat dan teknik agar serangan dapat

ISSN: 2303 – 2790
Bengkalis, 27 September 2023

diproses dan dikenali secepat mungkin. *DevOps* bukan hanya tentang tim pengembangan dan operasi. Jika Pemanfaatan ketangkasan dan daya tanggap aplikasi, tim keamanan juga harus memainkan peran terintegrasi dalam *full life cycle* pada aplikasi Anda (Red Hat, 2023).

Seiring berjalan waktu ancaman keamanan dan kerentanan dari suatu sistem semakin tinggi. Maka berdasarkan itu sangat diharapkan pengujian DevSecOps menggunakan metodologi *Static Application Security Testing* (SAST) dapat dilakukan, Teknik *Static Application Security Testing* (SAST) digunakan untuk memeriksa *source code* untuk mencari kerentanan yang dapat membahayakan keamanan dari software yang sedang dikembangkan. Ini menguntungkan karena semua kesalahan yang ditemukan selama pengujian dapat diatasi sebelum code dikompilasi (Ghazaly, 2021). salah satu platform yang dapat digunakan dalam hal pemindaian celah dan kesalahan adalah Snyk, Snyk merupakan sebuah platform keamanan siber yang dapat digunakan oleh para pengembang untuk melakukan pemindaian *code* program, Aplikasi berbasis *container* atau *cloud* dan dapat diintegrasikan dengan *IDE (Integrated Development Environment*).

METODE PENGUJIAN



Gambar 1 Topologi Jaringan

Metode pengujian pada proyek akhir ini adalah deployment. Pengujian dilakukan sebanyak 10 kali percobaan, Snyk melakukan *Structural analysis* yaitu memeriksa struktur *code* yang dilakukan secara real time. Snyk akan mengidentifikasi kelemahan dari struktur *code* tersebut serta memberikan klasifikasi terhadap kerentanan yang ditemukan, untuk menunjukkan risiko kerentanan tersebut dalam aplikasi. Terdapat

ISSN: 2303 – 2790
Bengkalis, 27 September 2023

beberapa klasifikasi yang telah ditentukan oleh snyk berdasarkan score dari CVSS *(Common Vulnerability Scoring System)* yaitu:

1. *Low Severity* (0.0-3.9)
   aplikasi dapat menampilkan beberapa data yang memungkinkan dapat digunakan dengan kerentanan lain untuk menyerang aplikasi.
2. *Medium Severity* (4.0-6.9)
   penyerang dalam kondisi tertentu dapat mengakses data sensitif pada aplikasi
3. *High Severity* (7.0-8.9)
   dapat mengizinkan penyerang untuk mengakses data sensitif pada aplikasi.
4. *Critical Severity* (9.0-10.0)
   memungkinkan penyerang mengakses data sensitif dan menjalankan *code* pada aplikasi.

Snyk juga menggunakan database yang menyimpan daftar kerentanan yang di laporkan dari pengguna dan tim snyk yang mengobservasi kerentanan dari CVSS dan OWASP dengan berbagai Aplikasi dan Sistem operasi yang dikenali oleh Snyk Security. Snyk menghitung skor berdasarkan sejumlah faktor, termasuk skor CVSS, tren kerentanan, keterjangkauan, ketersediaan exploit, dan faktor lainnya. Faktor-faktor ini memberikan skor dengan tingkat perincian yang tinggi Setelah Snyk memberikan daftar kerentanan yang telah berhasil diidentifikasi, Snyk akan memberikan saran untuk meminimalisir kelemahan dari struktur *code* pemrograman.

HASIL DAN PEMBAHASAN

1. Hasil dan Analisis Pengujian Kuantitatif

Pada penelitian ini, dilakukan pengujian terhadap 10 *website* yang ada di lingkungan Politeknik Caltex Riau. Tujuan dari pengujian ini adalah untuk mengumpulkan data kuantitatif terkait dengan tingkat keamanan dan kerentanan sistem yang digunakan oleh *website* tersebut. Melalui proses pengujian yang sistematis dan terstruktur, Snyk mengidentifikasi potensi celah keamanan yang ada pada masing-masing *website* dan mengumpulkan data terkait dengan tingkat kerentanan yang ditemukan. Hasil

ISSN: 2303 – 2790
Bengkalis, 27 September 2023

pengujian ini akan memberikan gambaran yang jelas tentang keadaan keamanan sistem website di lingkungan Politeknik Caltex Riau berdasarkan data kuantitatif yang terkumpul. Proses pengujian ini melibatkan serangkaian langkah yang meliputi analisis keamanan, identifikasi celah keamanan, serta pengumpulan data kuantitatif terkait dengan kerentanan sistem.

Setelah melakukan pengujian terhadap masing – masing fungsi dari sistem, hasil pengujian dapat dilihat pada Tabel 1

Tabel 1 Hasil Pengujian Terhadap Sistem

Nama Aplikasi	Base Aplikasi	Besar File (MegaByte)	Waktu Scanning (Detik)	Total Jumlah Kerentanan (Code Security)
Website utama kampus PCR	PHP 7.4	165 MB	00,35,43	90
Website CBT PCR	PHP 7.2	108 MB	01,19,60	54
Website billing PCR	PHP 7.4	187 MB	01,49,30	172
Website Sistem Akademik Orangtua	PHP 7.4	120 MB	01,32,99	51
Website PMB PCR	PHP 7.3	260 MB	01,39,66	107
Website Peminjaman Ruangan PCR	PHP 7.3	272 MB	01,50,80	167
Website Magister PCR	PHP 7.2	127 MB	01,26,73	90
Website PKM PCR	PHP 7.4	90.9 MB	01,29,64	72
Website Akademik Mahasiswa	PHP 7.3	881 MB	01,31,01	71
Website Peminjaman GSG PCR	PHP 7.4	169 MB	00,42,24	215
Total Kerentanan				1084

Hasil dari pengujian ini berupa data variable yang ditunjukkan snyk Ketika selesai melakukan scanning, Berdasarkan data pengujian kuantitatif yang diberikan, berikut adalah analisa yang dapat dilakukan:

1. Besar *File* dan Waktu *Scanning*: Terdapat variasi dalam ukuran *file* aplikasi dan waktu yang dibutuhkan untuk melakukan scanning.
2. Jumlah Kerentanan: Terdapat perbedaan dalam jumlah kerentanan yang ditemukan pada setiap aplikasi.
3. Distribusi Kerentanan: Terdapat variasi dalam distribusi kerentanan di antara aplikasi-aplikasi yang diuji. Beberapa aplikasi seperti "Website billing PCR" dan "Website peminjaman ruangan PCR" memiliki jumlah

ISSN: 2303 – 2790
Bengkalis, 27 September 2023

kerentanan yang cukup tinggi, sementara aplikasi lain seperti "Website utama kampus PCR" dan "Website Magister PCR" memiliki jumlah kerentanan yang lebih rendah

4. Fokus Perbaikan: Data ini memberikan petunjuk tentang area kerentanan yang perlu mendapatkan perhatian lebih.

Melalui analisis ini, *Programmer* dapat mengidentifikasi aplikasi-aplikasi yang memiliki tingkat kerentanan yang lebih tinggi dan menentukan prioritas perbaikan yang diperlukan. Selain itu, *Programmer* juga dapat mengevaluasi efektivitas langkah-langkah keamanan yang telah diimplementasikan dan mengambil tindakan untuk meningkatkan keamanan sistem atau aplikasi yang sedang dikembangkan.

2. Hasil dan Analisis Pengujian kuantitatif

Hasil wawancara menunjukkan bahwa para programmer sepenuhnya menyadari pentingnya mengamankan sistem/aplikasi untuk menjamin keaslian dan kehandalan sistem tersebut. Namun, tantangan utama yang dihadapi adalah terbatasnya sumber daya, sehingga waktu untuk melakukan pengujian yang mendalam menjadi terbatas, yang menyulitkan dalam mengidentifikasi potensi kelemahan sistem/aplikasi. Dalam konteks ini, peran alat dan teknologi terbukti krusial, membantu pengembang dalam mengidentifikasi celah keamanan dengan lebih cepat dan memberikan solusi perbaikan yang efisien. Para programmer telah menggunakan pendekatan seperti menciptakan function/helper yang telah teruji, serta menggunakan alat atau teknologi seperti SAST dan SNYK untuk mengidentifikasi dan meminimalisir celah keamanan.

Penggunaan alat dan teknologi tersebut telah memberikan dampak yang signifikan dalam mengurangi jumlah celah keamanan secara keseluruhan dalam sistem/aplikasi. Dalam upaya memastikan penggunaan yang optimal, para programmer melakukan pemeriksaan keamanan secara berkala pada sistem/aplikasi yang dikembangkan. Keberhasilan penggunaan alat dan teknologi dinilai dari seberapa banyak celah keamanan yang teridentifikasi dan langkah pencegahan serta perbaikan yang diambil. Penggunaan SAST dan SNYK juga berperan penting dalam memfasilitasi pemahaman tentang konsep dan prinsip keamanan dalam pengembangan sistem/aplikasi, membantu menjaga

ISSN: 2303 – 2790
Bengkalis, 27 September 2023

keamanan secara berkala. Oleh karena itu, penggunaan alat dan teknologi seperti SAST dan SNYK menjadi hal yang sangat diperlukan dalam memitigasi risiko keamanan dan mencegah kejadian keamanan yang dapat membahayakan sistem kampus.

SIMPULAN

Adapun kesimpulan yang dapat diambil mengenai implementasi proyek akhir ini adalah:

1. Hasil pengujian *scanning* menggunakan snyk dapat terlihat pada IDE Visual studio code dan *Dashboard* Snyk.io
2. Pengujian *workflow* CI/CD Github *Action* berjalan dengan baik, dan dapat dilakukan pada dua *branch* yang berbeda untuk melakukan *deployment container-public* dan *container-beta*.
3. Penggunaan Snyk sebagai alat pemindaian keamanan telah terbukti efektif dalam proyek ini, *Programmer* merasa bahwa dengan menggunakan Snyk SAST, pengembang dapat mendeteksi celah keamanan dalam *website* atau aplikasi sebelum diluncurkan ke publik.
4. Pemindaian menggunakan Snyk pada 10 website politeknik Caltex Riau menghasilkan temuan bahwa *website* peminjaman GSG mempunyai kerentanan terbanyak dan *website* utama PCR dan *Website* magister PCR menjadi *website* dengan kerentanan yang rendah.
5. Jumlah total kerentanan yang terdeteksi mencapai 1089.
6. Ada beberapa kerentanan kategori "*High*" yang terdeteksi, dengan jumlah terbanyak ditemukan pada "Website peminjaman GSG PCR" sebanyak 80 kerentanan
7. Dengan adanya hasil pengujian terhdap system, pengembang dapat memprioritaskan “Website peminjaman GSG” untuk pelaksanaan patching dan perbaikan keamanan.

DAFTAR PUSTAKA

Bijwe, A., & Shankar, P. (2022). Challenges of Adopting DevOps Culture on the Internet of Things Applications - A Solution Model. *2022 2nd International Conference on*

ISSN: 2303 – 2790
Bengkalis, 27 September 2023

Technological Advancements in Computational Sciences (ICTACS), 638–645. https://doi.org/10.1109/ICTACS56270.2022.9988182

Dharma, I. G. N. B. (2020). DevOps adoption in software engineering practices: A systematic literature review. *Journal of Software Engineering Research and Development*, *8*(1), 1–25. https://doi.org/https://doi.org/10.1186/s40411-020-00110-0

Kumar, V., & Rathore, R. S. (2018). Security Issues with Virtualization in Cloud Computing. *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 487–491. https://doi.org/10.1109/ICACCCN.2018.8748405

Mahdy Ghazaly, N. (2021). Learning the Idea Behind SAST (Static Application Security Testing) and How It Functions. *International Journal of Management and Engineering Research*, *1*(1), 1–4.

Red Hat. (2023). *What is DevSecOps?* https://www.redhat.com/en/topics/devops/what-is-devsecops

Seetharaman Jeganathan. (2019). DevSecOps: A Systemic Approach for Secure Software Development Securing Terminology: Lessons from Interdisciplinary Research Changing the DevOps Culture One Security Scan at a Time The Python Programming Language: Relational Databases Secure DevOps before. *ISSA Journal*, *17*(11), 20–37. https://issala.org/wp-content/uploads/2019/11/November19.pdf

Wang, F., Wang, H., & Xue, L. (2021). Research on Data Security in Big Data Cloud Computing Environment. *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 1446–1450. https://doi.org/10.1109/IAEAC50856.2021.9391025

Zunnurhain, K., & Duclervil, S. R. (2019). A New Project Management Tool Based on DevSecOps. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 239–243. https://doi.org/10.1109/CSCI49370.2019.00049

ISSN: 2303 – 2790
Bengkalis, 27 September 2023