Jurnal Pendidikan dan Teknologi Indonesia (JPTI)
Vol. 5, No. 11, November 2025, Hal. 3420-3434
DOI: https://doi.org/10.52436/1.jpti.1126
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Homelab DevSecOps: Perancangan Kerangka Kerja Pengembangan Aplikasi yang Selaras dengan Pelindungan Data Pribadi pada Lingkungan Komputasi Terbatas

Geraldo Martua Sigalingging[*1]

[1]Fakultas Teknik, Universitas Indonesia, Indonesia
Email: [1]geraldo.martua31@ui.ac.id

Abstrak

Perkembangan teknologi aplikasi menuntut kecepatan dan efisiensi, mendorong adopsi *DevOps* sebagai model pengembangan aplikasi kolaboratif. Namun, inovasi ini beriringan dengan meningkatnya ancaman siber dan risiko kebocoran data pribadi, memicu perkembangan ke *DevSecOps* yang mengintegrasikan keamanan di seluruh siklus pengembangan aplikasi. Penelitian ini berupaya mengatasi tantangan tersebut dengan merancang model alur DevSecOps yang mematuhi regulasi perlindungan data pribadi, khususnya **GDPR** dan **UU PDP Indonesia**, dalam lingkungan komputasi terbatas (*homelab*). Metodologi penelitian meliputi lima tahap yaitu, studi literatur, perancangan, implementasi, pengujian dan validasi, serta penarikan kesimpulan. Berdasarkan analisa studi literatur, dihasilkan model alur DevSecOps dengan tujuh fase yaitu, Plan, Code, Build, Test, Deploy, Operate, dan Monitor. Setiap fase diperkuat dengan aktivitas keamanan yaitu Threat Modeling, Static Application Security Testing (SAST), Container Scanning, Dynamic Application Security Testing (DAST), Compliance Assessment, Vulnerability Assessment, serta File Integrity Monitoring (FIM) & Security Information and Event Management (SIEM). Aktivitas didukung perangkat selaras regulasi pelindungan data pribadi. Pengujian dilakukan pada tiga aplikasi dengan kerentanan keamanan yang disengaja (OWASP Juice Shop, DVWA, DVJA). Hasil pengujian berhasil mengidentifikasi celah keamanan seperti model ancaman, kerentanan kode, infrastruktur dan kontainer, dan ketidakpatuhan regulasi. Temuan ini menyediakan umpan balik bagi tim pengembang, keamanan, dan operasional untuk perbaikan berkelanjutan. Penelitian ini berkontribusi menyediakan model alur DevSecOps yang teruji, lengkap, relevan, dan terbukti dapat diimplementasikan dalam lingkungan komputasi terbatas. Model DevSecOps ini meningkatkan keamanan aplikasi sekaligus memastikan kepatuhan regulasi perlindungan data yang berlaku yang panduan penting bagi pengembangan aplikasi yang aman.

Kata kunci: *DevSecOps, GDPR, HomeLab, UU PDP*

Homelab DevSecOps: Designing an Application Development Framework Aligned with Personal Data Protection in Resource-Constrained Computing Environments

Abstract

The rapid advancement in application technology demands speed and efficiency, driving the adoption of DevOps as a collaborative development model. This innovation is paralleled by increasing cyber threats and personal data breaches, pushing the evolution to DevSecOps, which integrates security across the entire development lifecycle. This research addresses these critical challenges by designing a DevSecOps pipeline model that adheres to personal data protection regulations, specifically GDPR and the Indonesian UU PDP, within a resource-constrained computing environment (homelab). Our methodology involved five key stages: a literature review, designing, implementation, testing and validation, and conclusive. Analyzing referenced pipelines, we developed a robust seven-phase DevSecOps model: Plan, Code, Build, Test, Deploy, Operate, and Monitor. Each phase incorporates specific security activities: Threat Modeling, SAST/DAST, Compliance Assessment, Vulnerability Assessment, and File Integrity Monitoring (FIM) & Security Information and Event Management (SIEM). All activities are supported by tools aligned with data protection regulations. Testing was performed on three vulnerable applications (OWASP Juice Shop, DVWA, DVJA). The testing successfully identified various security gaps, including threat models, code vulnerabilities, infrastructure and container weaknesses, and regulatory non-compliance. These findings provide feedback for development, security, and operations teams, enabling continuous improvement. This research contributes by delivering a, comprehensive, relevant, and implementable DevSecOps pipeline model for resource-constrained environments. This model strengthens application security and ensures compliance with data protection regulations, serving as a guide for secure application development.

Keywords: *DevSecOps, GDPR, HomeLab, UU PDP*

p-ISSN: 2775-4227
e-ISSN: 2775-4219

1. PENDAHULUAN

Perkembangan teknologi aplikasi sangat pesat dalam beberapa waktu terakhir. Kecepatan dan efisiensi waktu menjadi faktor utama yang memengaruhi keberhasilan pengembangan aplikasi. Faktor ini juga mengevolusi model alur dalam pengembangan aplikasi, dimulai dari model alur linear seperti waterfall, berkembang ke model alur yang dinamis seperti Agile dan saat ini bergerak menuju model alur yang fokus pada kolaborasi tim pengembang aplikasi seperti Scrum dan Kanban [1].

Kolaborasi ini juga berkembang dimana sebelumnya kolaborasi hanya pada tim pengembang, sekarang diperlukan juga kolaborasi dengan tim operasional yang berperan dalam melakukan pengujian dan publikasi aplikasi. Kolaborasi ini menciptakan model alur pengembangan aplikasi baru yang dinamakan DevOps (Developer Operations) yang menghapus sekat diantara tim pengembang dan tim operasional [2]. DevOps memungkinkan pengembangan aplikasi yang lebih cepat, pengujian aplikasi yang lebih komperhensif, dan menciptakan mekanisme umpan balik yang berkelanjutan, sehingga pengembangan perangkat lunak dapat dilakukan secara iteratif dan responsif terhadap perubahan kebutuhan [3].

Walaupun sudah memakai model alur yang kokoh, pengembangan aplikasi tetap mempunyai ancaman keamanan seperti serangan siber atau kebocoran data. Menurut data dari Surfshark, dari tahun 2004 – 2025 terdapat lebih dari 177 juta kasus kebocoran data pribadi di Indonesia yang setara 0,9% dari total kebocoran data secara global [4]. Data ini menunjukkan tingginya risiko keamanan informasi dan meningkatkan kesadaran tim pengembang dan tim operasional akan pentingnya keamanan dalam pengembangan aplikasi sehingga terjadi evolusi DevOps menjadi DevSecOps (Developer, Security, dan Operations) [5]. DevSecOps mengenalkan pendekatan "Shift Left" dimana proses pengujian dan penerapan keamanan yang sebelumnya dilakukan di akhir menjelang aplikasi dipublikasikan digeser ke tahap awal pengembangan aplikasi sehingga seluruh tahapan pengembangan aplikasi akan terintegrasi standar keamanan [6].

DevSecOps juga membantu organisasi dalam memenuhi berbagai persyaratan kepatuhan (compliance) terhadap standar keamanan dan regulasi yang berlaku [7]. Salah satu kepatuhan (compliance) yang penting dalam penerapan DevSecOps adalah pelindungan data pribadi. Menurut hasil laporan IBM yang berjudul "Cost of a Data Breach Report 2023" menunjukkan DevSecOps bermanfaat meminimalisir dampak biaya kerugian akibat pembobolan data karena DevSecOps memastikan bahwa seluruh proses pengembangan dan operasional sistem dilengkapi dengan mekanisme pelindungan data pribadi yang kuat dan sesuai dengan regulasi yang berlaku [8].

Beberapa regulasi telah diberlakukan dan salah satu regulasi yang paling komprehensif adalah General Data Protection Regulation (GDPR) yang diterbitkan oleh Uni Eropa. GDPR menetapkan standar terkait pengumpulan, penyimpanan, pengolahan, dan distribusi data pribadi, serta menuntut organisasi untuk bertanggung jawab secara penuh terhadap perlindungan informasi pribadi pengguna [9]. Indonesia juga mengesahkan Undang-Undang Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi (UU PDP). UU PDP menjadi instrumen hukum pertama di Indonesia yang secara khusus mengatur tentang tata kelola data pribadi, baik yang dikumpulkan oleh sektor publik maupun swasta [10]. Dr. Sinta Dewi Rosadi, S.H., LL.M. dalam tulisannya mengenai UU PDP Indonesia, menjelaskan bahwa meskipun GDPR dan UU PDP sama-sama bertujuan utama untuk melindungi data pribadi, tetapi keduanya memiliki beberapa perbedaan prinsip [10]. Perbedaan-perbedaan ini, seperti yang tertera dalam Tabel 1, mencakup beberapa perbedaan aspek ruang lingkup penerapan aturan, hak-hak yang dimiliki oleh subjek data, dan kewajiban-kewajiban bagi pengendali data.

Tabel 1. Perbedaan Prinsip Dasar GDPR dan UU PDP

GDPR (2018)	UU PDP (2022)
Keabsahan, Keadilan, dan Transparansi	Terbatas, Spesifik, Sah, dan Transparan
Pembatasan Tujuan	Sesuai Tujuan
Ketepatan dan Akurasi	Jaminan atas Hak Subjek Data Pribadi
Batasan Penyimpanan	Akurat, Lengkap, Tidak Menyesatkan, dan Dapat Dipertanggungjawabkan
Integritas dan Kerahasiaan	Aman dari Tindakan Tidak Sah, Penyalahgunaan, Perusakan, Penghilangan
Akuntabilitas	Pemberitahuan Tujuan, Aktivitas, dan Kegagalan Pelindungan
—	Pemusnahan pada Akhir Masa Penyimpanan atau Berdasarkan Permintaan
—	Pertanggungjawaban dan Pembuktian

Penerapan DevSecOps menghadirkan sejumlah tantangan, terutama karena mayoritas tim pengembang masih cenderung menyerahkan tanggung jawab keamanan kepada tim keamanan khusus. Selain itu, investasi dalam keamanan sistem sering dipandang mahal, tidak memberikan keuntungan langsung, dan tidak menjamin perlindungan mutlak [11]. Menjawab tantangan ini, penelitian ini bertujuan untuk merancang kerangka kerja

p-ISSN: 2775-4227
e-ISSN: 2775-4219

model alur DevSecOps yang tidak hanya mematuhi regulasi perlindungan data pribadi, tetapi juga dapat diimplementasikan dalam lingkungan komputasi terbatas.

Kebaruan penelitian ini terletak pada penyediaan tahapan, aktivitas, dan perangkat yang terperinci dalam model alur DevSecOps yang patuh terhadap regulasi pelindungan data pribadi. Penelitian ini juga mengimplementasikan model alur DevSecOps tersebut dalam lingkungan komputasi terbatas seperti HomeLab. Penelitian ini memberikan kontribusi signifikan karena penelitian sebelumnya yang direferensikan dalam studi ini belum secara eksplisit membahas pemilihan alur, aktivitas, dan perangkat DevSecOps yang tepat, serta belum secara lengkap mengimplementasikan alur, aktivitas, dan perangkat DevSecOps ke dalam sistem, serta belum ada penelitian yang mengimplementasikannya dalam ruang lingkup komputasi terbatas seperti HomeLab.

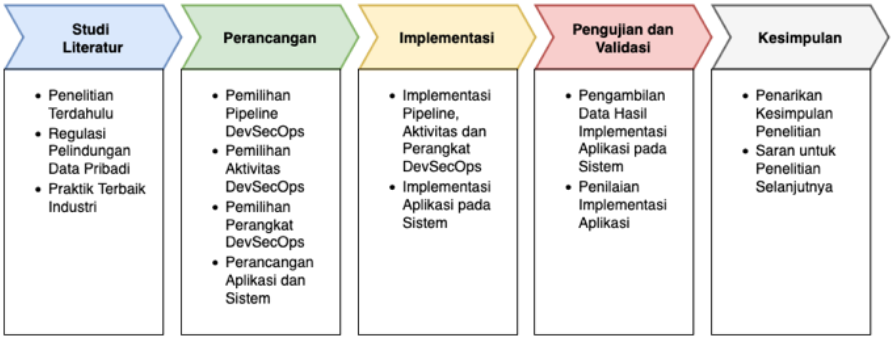
2. METODE PENELITIAN

Metodologi penelitian ini disusun melalui 5 tahapan yang sistematis seperti yang tertera pada gambar 1. Tahap awal adalah studi literatur yang mencakup tinjauan pustaka dari tiga sumber literatur. Literatur pertama adalah analisis terhadap regulasi perlindungan data dengan fokus pada General Data Protection Regulation (GDPR) Uni Eropa dan Undang-Undang Perlindungan Data Pribadi (UU PDP) Indonesia, literatur kedua adalah kajian terhadap penelitian-penelitian terdahulu yang relevan, dan literatur ketiga adalah identifikasi praktik terbaik (best practices) dalam implementasi DevSecOps dan keamanan data.

Berlandaskan studi literatur, penelitian dilanjutkan dengan tahap kedua yaitu perancangan (design). Proses ini meliputi penentuan fase DevSecOps yang akan diadopsi, penetapan aktivitas dalam setiap fase, pemilihan perangkat (tools) pendukung, serta perancangan arsitektur sistem yang akan menjadi objek pengujian. Tahap ketiga adalah implementasi, di mana seluruh elemen yang telah direncanakan akan diimplementasikan pada sistem yang juga telah dirancang.

Setelah sistem berhasil diimplementasikan dilanjutkan ke tahap keempat yaitu dan validasi untuk mengukur efektivitas dan kesesuaian sistem. Pengujian ini dilakukan dengan mengambil dan menganalisis data hasil implementasi, yang diukur menggunakan metrik keamanan seperti jumlah vulnerability dan tingkat keparahan vulnerability berdasarkan skor CVSS. Hasil dari pengujian dan validasi ini kemudian dianalisis.

Hasil dari pengujian dan validasi ini kemudian dianalisis dan dipaparkan secara rinci. Tahap kelima merupakan penutup, metodologi ini diakhiri dengan penarikan kesimpulan berdasarkan keseluruhan temuan penelitian, serta evaluasi kritis untuk menentukan apakah hasil yang diperoleh telah berhasil menjawab tujuan penelitian yang telah dirumuskan pada bagian awal.



Gambar 1. Metodologi Penelitian

3. PERANCANGAN SISTEM

3.1 Penentuan Pipeline DevSecOps

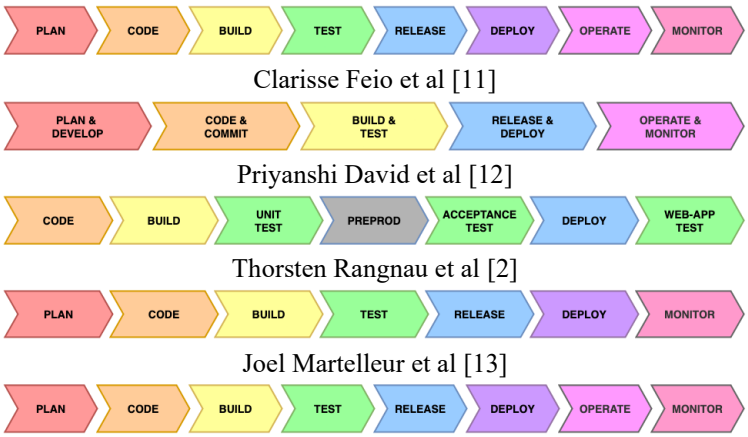
Komponen fundamental dalam kerangka kerja DevSecOps adalah pipeline yaitu serangkaian tahapan terstruktur yang dilalui oleh sebuah proses pengembangan aplikasi sebelum dapat dipublikasikan atau dioperasikan. Pipeline ini memiliki banyak variasi yang dapat disesuaikan dengan kebutuhan penggunaan dan fungsi masing-masing proyek. Penentuan pipeline yang tepat sangat penting, terutama dalam konteks pemrosesan data pribadi, sebagaimana diatur dalam Pasal 28 ayat (1) UU PDP menyatakan bahwa "Pengendali Data Pribadi wajib melakukan pemrosesan Data Pribadi sesuai dengan tujuan pemrosesan Data Pribadi." sehingga, seluruh proses DevSecOps harus memiliki tahapan yang jelas dan terukur dalam pengelolaan data pribadi.

Penelitian ini fokus pada perancangan dan analisis pipeline DevSecOps dan kriteria utama dalam pemilihan dan perancangan pipeline ini adalah sebagai berikut:

p-ISSN: 2775-4227
e-ISSN: 2775-4219

- Pipeline harus selaras dengan prinsip DevSecOps namun tetap ringan dalam penggunaan sumber daya komputasi (CPU, memori, penyimpanan).
- Pipeline dirancang agar mudah diimplementasikan dan dikelola dalam sistem berskala kecil atau homelab, menghindari kompleksitas yang tidak perlu.
- Pipeline tidak boleh terkunci pada satu ekosistem sumber sehingga dapat memberikan fleksibilitas adopsi yang lebih luas.

Melalui analisis tersebut, aspek-aspek umum dari berbagai pipeline dapat diidentifikasi guna menetapkan fase-fase yang akan dipilih dalam DevSecOps ini. Sebanyak lima pipeline telah dipilih untuk dipelajari dan dibandingkan. Setiap pipeline referensi dijajarkan secara berdampingan untuk memudahkan perbandingan visual. Setiap fase pada masing-masing pipeline diberi penandaan warna yang konsisten sesuai dengan kategori atau fungsinya dalam siklus pengembangan perangkat lunak. Hasil dari komparasi visual kelima pipeline referensi ini dapat dilihat secara detail pada Gambar 2.



Gambar 2. Komparasi Perbandingan Pipeline DevSecOps pada Penelitian Sebelumnya

Dari perbandingan ini, teridentifikasi bahwa meskipun terdapat variasi antar kelima pipeline yang dianalisis, terdapat similaritas yang signifikan pada level proses inti.

Pipeline dari Clarisse Feio et al. [11] menunjukkan keunggulan dalam aspek kelengkapan dokumentasi. Pipeline ini menyajikan tahapan secara terstruktur dan jelas, sehingga memudahkan proses implementasi serta menjadikannya referensi yang kuat dalam konteks akademik dan praktis. Pipeline ini memiliki tingkat detail yang tepat tanpa membebani pengguna dengan kompleksitas yang berlebihan.

Pipeline dari Priyanshi David et al. [12] menunjukkan model ini menonjol dalam hal simplisitas. Pendekatan penggabungan dua fase yang memiliki kemiripan berhasil menyederhanakan struktur pipeline dan mempermudah kategorisasi aktivitas di dalamnya. Kelemahan dari pipeline ini adalah konsentrasi aktivitas yang lebih banyak dalam satu fase gabungan, yang berpotensi meningkatkan kompleksitas atau durasi eksekusi fase tersebut dibandingkan dengan model yang memisahkan fase-fase serupa.

Pipeline dari Thorsten Rangnau et al. [2], menunjukkan kelebihan signifikan dengan fokusnya yang kuat pada aspek testing. Hal ini menjadikan pipeline yang dapat menjadi referensi untuk merancang dan mengimplementasikan aktivitas pengujian keamanan. Namun, kekurangan pipeline ini adalah absennya tahapan pasca-deployment, sehingga kurang memenuhi aspek kontrol dan monitoring berkelanjutan.

Pipeline dari Joel Martelleur et al. [13] ini memiliki kemiripan struktural dengan pipeline sebelumnya. Kelemahan dari penelitian ini adalah penelitian hanya berfokus fokus dominan proses pengembangan aplikasi saja.

Pipeline dari Wen-Tin Lee et al.[14] juga memiliki kemiripan struktural dengan pipeline sebelumnya. Kelebihan referensi ini adalah penelitian telah mencakup implementasi praktis dari pipeline ini, memberikan bukti konsep yang memperkuat validitas pipeline dalam konteks aplikatif dan kematangan konsep yang telah terbukti.

Dari seluruh pipeline yang dianalisis, pipeline yang dikemukakan oleh Clarisse Feio et al. [11] dan Wen-Tin Lee et al. [14] memberikan kontribusi signifikan terhadap penelitian ini. Keduanya memberikan keseimbangan yang optimal antara kelengkapan dan kesederhanaan, tanpa menimbulkan kompleksitas berlebihan. Pipeline yang dirancang untuk penelitian ini mengadopsi ketujuh fase tersebut (Plan, Code, Build, Test, Deploy, Operate, Monitor) yang dapat dilihat pada gambar 3.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Gambar 3. Pipeline DevSecOps yang Ditentukan untuk Penelitian Ini.

Arsitektur ini cukup mempresentasikan setiap langkah utama dalam siklus DevSecOps, sehingga detail penting tidak tersembunyi dalam fase yang terlalu umum. Di sisi lain, pipeline ini juga menjaga tingkat kesederhanaan yang memadai agar tidak terasa terlalu kompleks untuk diimplementasikan dan dikelola dalam lingkungan Homelab.

3.2 Penentuan Aktivitas DevSecOps

Setelah menentukan pipeline yang sesuai, langkah selanjutnya adalah menetapkan aktivitas spesifik untuk setiap fase dalam pipeline DevSecOps tersebut. Proses penentuan aktivitas ini disesuaikan dengan Pasal 35 UU PDP yang menyatakan bahwa pengendali data pribadi wajib melindungi dan memastikan keamanan data dengan menerapkan langkah teknis operasional. Pasal tersebut menekankan kewajiban Pengendali Data Pribadi untuk "melindungi dan memastikan keamanan Data Pribadi yang diprosesnya dengan menyusun dan menerapkan langkah teknis operasional untuk mencegah pelanggaran Pelindungan Data Pribadi".

Dalam penentuan aktifitas yang dilakukan dalam model alur DevSecOps yang regulasi pelindungan data pribadi serta diimplementasikan di lingkungan komputasi terbatas sehingga untuk melakukan penentuan aktivitas DevSecOps perlu memenuhi kriteria:

- Setiap aktivitas DevSecOps yang dipilih harus memiliki keterkaitan langsung atau mendukung pelaksanaan implementasi atau pengujian keamanan
- Setiap fase dalam pipeline yang telah ditetapkan harus mencakup setidaknya satu aktivitas yang secara eksplisit terkait dengan implementasi atau pengujian keamanan
- Setiap aktivitas yang dipilih terbatas dan berfokus pada keamanan aplikasi dan keamanan infrastruktur
- Setiap aktivitas diimplementasikan dalam lingkungan komputasi terbatas dan tidak kompleks.

Untuk mengidentifikasi aktivitas DevSecOps yang sesuai, diterapkan pendekatan dua tahap. Tahap pertama melibatkan analisis literatur yang komprehensif untuk mengidentifikasi aktivitas yang banyak dilakukan di sepanjang berbagai alur kerja DevSecOps yang diusulkan oleh para peneliti dan praktisi. Tahap kedua, daftar aktivitas umum yang telah teridentifikasi dan memenuhi seluruh persyaratan kriteria yang telah ditetapkan sebelumnya kemudian diselidiki dan dievaluasi lebih lanjut. Melalui tinjauan literatur dari penelitian-penelitian sebelumnya, berbagai aktivitas berhasil diidentifikasi. Daftar aktivitas yang terkumpul ini kemudian dicatat dan dikelompokkan menurut tahapan-tahapan pipeline DevSecOps sebelumnya sebelumnya.

Hasil dari pengelompokan ini dituliskan dalam tabel 2, yang berisi rincian aktivitas untuk setiap fase. Tabel tersebut menggambarkan praktik-praktik standar yang umum diterapkan dalam proses DevSecOps.

Tabel 2. Kumulatif Perbandingan Aktivitas DevSecOps pada Penelitian Sebelumnya

Fase	Aktivitas
Plan	Project & Security Planning, Requirement Analysis, Threat Modeling, Risk Assessment, Compliance Check, Secure Coding Preparation
Code	Secure Coding Practices, Code Review, SAST (Static Application Security Testing), Version Control Management, SCA (Software Composition Analysis), Secret Management
Build	Build Automation, Dependency Analysis, IaC Security, Container Scanning, Patch Management, Artifact Management, Infrastructure Scanning
Test	Integration Testing, UAT, DAST, IAST, HAST, Penetration Testing, Infrastructure Testing, RASP, Vulnerability Scanning, Container & Cloud Security Testing, SIEM Integration
Deploy	Infrastructure Hardening, Container Scan, Vulnerability Assessment, QA Test, Performance Test, IDS/IPS, Compliance Assessment, SIEM Integration, Artifact & Secret Management
Operate	Patching and Updates, Logging, Threat Intelligence, Vulnerability Assessment, Incident Management, Red & Blue Team Exercises, Bug Bounty Programs, SIEM Integration
Monitor	Performance Monitoring, File Integrity Monitoring, Logging & Log Analysis, Threat Intelligence, IDS/IPS, WAF, SIEM Integration, VAPT, Incident Management

Selaras dengan UU PDP, khususnya pasal 28, 31, dan 35, setiap tahapan dalam pemrosesan data pribadi dari fase perencanaan hingga operasional diwajibkan dilakukan secara sah, adil, dan dapat dipertanggungjawabkan, dalam penelitian ini, setiap fase dalam pipeline DevSecOps dipilih hingga mengandung minimal satu aktivitas yang secara eksplisit berorientasi pada keamanan data dan kepatuhan terhadap UU PDP. Setelah analisis, telah

p-ISSN: 2775-4227
e-ISSN: 2775-4219

dipilih sejumlah aktivitas yang akan dilaksanakan dalam proses DevSecOps ini. Aktivitas-aktivitas tersebut dapat dilihat pada tabel 3.

Tabel 3. Aktivitas DevSecOps yang Ditentukan untuk Penelitian Ini.

Fase	Aktivitas
Plan	Threat Modeling
Code	SAST (Static Application Security Testing)
Build	Container Scanning
Test	DAST (Dynamic Application Security Testing)
Deploy	Compliance Assessment
Operate	Vulnerability Assessment
Monitor	File Integrity Monitoring, SIEM Integration

Aktivitas yang ditentukan tidak hanya memenuhi standar DevSecOps dari sisi teknis, tetapi juga selaras dengan kebutuhan regulasi nasional dalam perlindungan data pribadi, sehingga pipeline yang dirancang dapat memberikan jaminan keamanan, transparansi, dan kepatuhan terhadap kebijakan yang berlaku.

3.3 Penentuan Perangkat DevSecOps

Setelah menetapkan aktivitas yang sesuai untuk setiap fase dalam DevSecOps, langkah selanjutnya dalam metodologi penelitian ini adalah menentukan perangkat (Tools) yang akan digunakan untuk mendukung setiap aktivitas tersebut. Dalam UU PDP, khususnya Pasal 22, 29, 31, dan 36, menekankan pentingnya pelindungan data pribadi secara komprehensif di seluruh siklus hidupnya mulai dari tahap pengumpulan, penyimpanan, pemrosesan, hingga penghapusan atau pemusnahan. UU PDP juga mensyaratkan implementasi langkah-langkah keamanan teknis, organisasional yang memadai, serta prinsip traceability (ketertelusuran) dan auditability (keterauditan) menjadi penting untuk memastikan setiap aktivitas pemrosesan data dapat ditelusuri dan dipertanggungjawabkan.

Penentuan perangkat dalam DevSecOps perlu memenuhi serangkaian kriteria spesifik, yang dirancang untuk meminimalkan potensi penundaan implementasi dan biaya operasional, kriteria tersebut adalah

- Prioritas perangkat yang bersifat OSS untuk meminimalkan biaya implementasi, memberikan fleksibilitas kustomisasi, serta mengurangi risiko ketergantungan pada ekosistem aplikasi tertentu
- Perangkat yang dipilih harus dalam tahap pengembangan aktif, dengan pembaruan terakhir tidak lebih dari satu tahun, memiliki dukungan komersil atau komunitas dan dokumentasi yang solid dan lengkap.
- Perangkat dapat menunjukkan kepatuhan dengan regulasi pelindungan data seperti GDPR dan UU PDP, dan/atau direkomendasikan oleh organisasi terpercaya di bidang keamanan siber dan teknologi.
- Perangkat harus diimplementasikan secara mandiri dalam infrastruktur pengguna untuk kontrol penuh atas data. Kemudahan proses instalasi, konfigurasi, dan operasional menjadi pertimbangan penting.

Penelitian ini berhasil mengidentifikasi 90 perangkat yang dirujuk dari studi sebelumnya. Melalui proses pemilihan dan berdasarkan kriteria yang sudah ditentukan, telah terpilih 13 perangkat (tools) DevSecOps. Perangkat yang tertera pada tabel 4 dinilai paling sesuai untuk diterapkan dalam kerangka kerja DevSecOps.

Tabel 4. Perangkat DevSecOps yang Ditentukan untuk Penelitian Ini.

Fase	Aktivitas	Perangkat
Plan	Threat Modeling	Threagile
Code	SAST (Static Application Security Testing)	SonarQube
Build	Container Scanning	Trivy
Test	DAST (Dynamic Application Security Testing)	OWASP PTK
Deploy	Compliance Assessment	Wazuh Compliance Monitoring
Operate	Vulnerability Assessment	Nessus Scan
Monitor	File Integrity Monitoring	Wazuh FIM
	SIEM Integration	Wazuh SIEM

Aktivitas yang ditentukan tidak hanya memenuhi standar DevSecOps dari sisi teknis, tetapi juga selaras dengan kebutuhan regulasi nasional dalam perlindungan data pribadi, sehingga pipeline yang dirancang dapat memberikan jaminan keamanan, transparansi, dan kepatuhan terhadap kebijakan yang berlaku.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

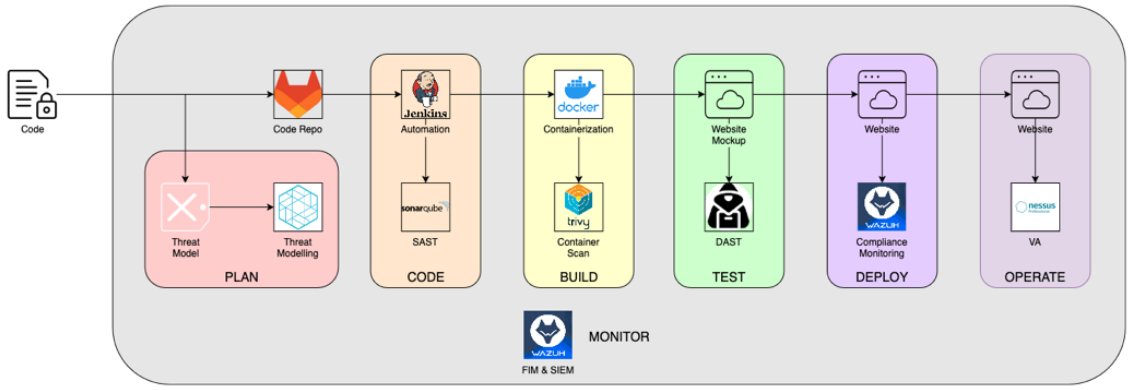
3.4 Perancangan Topologi Sistem DevSecOps

Setelah menentukan pipeline, aktivitas dan perangkat yang tepat untuk setiap fase dalam DevSecOps, tahap selanjutnya dalam metodologi penelitian ini adalah merancang dan menerapkan implementasi DevSecOps yang dibuat dalam sebuah topologi kerja DevSecOps. Sesuai dengan penerapan kerangka kerja yang mematuhi regulasi pelindungan data pribadi maka implementasi sistem dalam penelitian ini berlandaskan pada empat prinsip utama, yaitu: security by default, privacy by design, monitoring & auditability, serta incident response preparedness

Untuk mencapai implementasi yang sistematis dan sesuai dengan prinsip-prinsip tersebut, penelitian ini membaginya ke dalam tiga komponen utama yaitu:

- Topologi arsitektur dari lingkungan homelab, termasuk penempatan komponen dan alur komunikasi.
- Rincian perangkat keras yang digunakan untuk membangun lingkungan homelab.
- Rincian karakteristik aplikasi yang berfungsi sebagai objek pengujian dalam kerangka kerja DevSecOps

Dengan memperhatikan kriteria yang sudah ditentukan, rancangan topologi arsitektur telah digambarkan dan disajikan pada Gambar 4.



Gambar 4. Topologi Arsitektur Sistem DevSecOps

Topologi arsitektur yang telah dirancang akan diimplementasikan pada lingkungan homelab dengan konfigurasi perangkat keras dan perangkat lunak sebagai berikut:

- CPU: 14 Core
- RAM: 32 GB
- Sistem Operasi: Rocky Linux 9
- Aplikasi Pendukung: Docker

Dalam arsitektur penelitian ini, pemilihan objek pengujian menjadi tahap krusial, di mana tiga aplikasi spesifik dimanfaatkan sebagai subjek dalam kerangka kerja DevSecOps: OWASP Juice Shop, DVWA, dan DVJA. Ketiga aplikasi ini dipilih karena statusnya sebagai intentionally insecure web applications (aplikasi web yang sengaja dibuat tidak aman), yang menyediakan lingkungan terkontrol untuk validasi praktik keamanan. OWASP Juice Shop, sebagai proyek yang dikelola oleh OWASP, merepresentasikan aplikasi web modern yang realistis untuk simulasi serangan siber dan pengujian teknik pertahanan.

Melengkapi Juice Shop, digunakan pula DVWA (Damn Vulnerable Web Application), yaitu sebuah aplikasi berbasis PHP/MySQL yang dirancang dengan berbagai kerentanan umum untuk menjadi sarana pelatihan bagi para praktisi keamanan siber. Selanjutnya, DVJA (Damn Vulnerable Java Application) diaplikasikan sebagai representasi aplikasi yang dibangun dengan teknologi Java, juga dengan kerentanan yang disengaja, untuk memastikan cakupan pengujian keamanan pada arsitektur perangkat lunak yang beragam.

Ketiga aplikasi ini diimplementasikan dalam lingkungan terisolasi menggunakan teknologi kontainerisasi Docker untuk menjamin konsistensi dan portabilitas. Orkestrasi alur kerja DevSecOps dijalankan melalui Jenkins, yang secara otomatis memicu serangkaian pemindaian keamanan terintegrasi, meliputi Static Application Security Testing (SAST), Container Scanning, dan Dynamic Application Security Testing (DAST). Sebagai lapisan keamanan tambahan, ketiga server tersebut dievaluasi secara periodik melalui Vulnerability Assessment (VA) menggunakan Nessus dan diintegrasikan dengan platform Wazuh untuk kapabilitas Security Information and Event Management (SIEM) serta File Integrity Monitoring (FIM) secara waktu-nyata.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

4. IMPLEMENTASI SISTEM

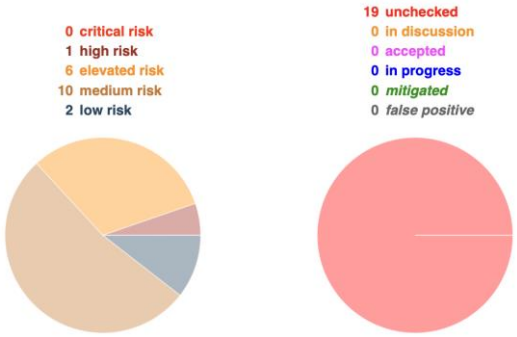
Setelah menentukan pipeline, aktivitas, perangkat serta topologi sistem yang tepat untuk setiap fase dalam DevSecOps, langkah berikutnya dalam metodologi penelitian ini adalah melakukan pengujian dan verifikasi terhadap kerangka kerja DevSecOps yang telah dirancang. Pengujian ini bertujuan untuk mengevaluasi efektivitas dan efisiensi implementasi yang dilakukan, serta untuk membuktikan penerapan prinsip dan ketentuan dalam UU PDP Indonesia dapat meningkatkan keamanan dan tata kelola pelindungan data pribadi.

Untuk mendukung pembuktian ini, penelitian dilakukan dengan menjalankan implementasi, mengambil data hasil implementasi dan menganalisa hasil implementasi. Hasil implementasi diukur menggunakan metrik keamanan yaitu:

- Jumlah Vulnerability: Jumlah kerentanan keamanan yang terdeteksi pada aplikasi
- Severity of Vulnerabilities: Analisis skor CVSS (Common Vulnerability Scoring System) dari kerentanan yang ditemukan, diklasifikasikan dengan tingkat keparahan (critical, high, medium, low).
- Log Results: Hasil untuk melihat hasil log saat proses DevSecOps berjalan.

4.1 Hasil Pengujian pada Tahap Plan

Pada tahap Rencana (Plan) dalam siklus DevSecOps, fokus utama adalah identifikasi terhadap potensi ancaman keamanan melalui implementasi pemodelan ancaman (threat modeling). Dalam penelitian ini, pemodelan ancaman menggunakan Threagile dan dijalankan secara sistematis terhadap arsitektur aplikasi. Proses ini diawali dengan pendefinisian parameter aplikasi, seperti fungsi-fungsi utama, serta klasifikasi aset data berdasarkan atribut penting termasuk tingkat kerahasiaan, integritas, ketersediaan, dan tingkat kepentingannya. Atribut-atribut ini kemudian didokumentasikan secara terstruktur ke dalam berkas berformat YAML yang merepresentasikan model ancaman dari aplikasi OWASP Juice Shop.



Gambar 5. Hasil Threat Modeling Menggunakan Threagile

File YAML berfungsi sebagai model masukan untuk dianalisis oleh Threagile. Setelah Threagile memproses berkas YAML tersebut, dihasilkan sebuah laporan analisis risiko. Laporan yang tertera pada gambar 5 berisi informasi hasil identifikasi dan kategorisasi potensi risiko keamanan ke dalam beberapa tingkatan berdasarkan dampaknya, meliputi Kritis (Critical), Tinggi (High), Sedang (Medium), dan Rendah (Low). Hasil dari Threat Modeling aplikasi OWASP Juice Shop tertera pada tabel 5, yaitu

Tabel 5. Hasil Threat Modeling Menggunakan Threagile

Kritikalitas	Kritis	Tinggi	Sedang-Tinggi	Sedang	Rendah
Jumlah	0	1	6	10	2

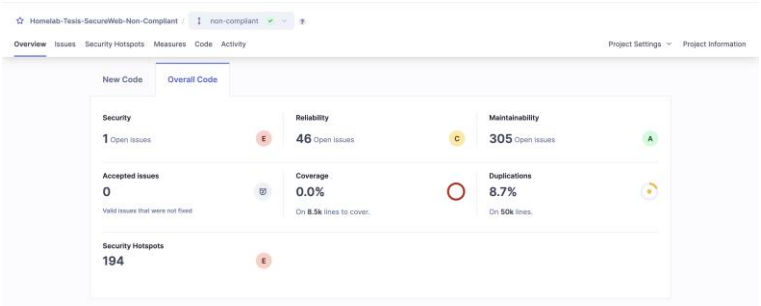
Laporan ini memberikan data mendetail mengenai jenis ancaman dan bagaimana melakukan remediasi terhadap ancaman tersebut yang kemudian dapat digunakan untuk memperbaiki keamanan aplikasi sebelum memasuki tahap pengembangan lebih lanjut.

4.2 Hasil Pengujian pada Tahap Code

Setelah tahap Perencanaan (Plan) yang berfokus pada pemodelan ancaman, siklus DevSecOps berlanjut ke tahap Kode (Code). Pada fase ini, penekanan diberikan pada analisis keamanan langsung terhadap kode sumber aplikasi. Aktivitas yang dilaksanakan adalah Static Application Security Testing (SAST).

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Dalam proses implementasi SAST, penelitian ini memanfaatkan platform SonarQube. Mekanisme kerja SonarQube melibatkan penarikan kode sumber aplikasi secara langsung dari repositori kode (code repository) yang digunakan oleh tim pengembang. Code Repository yang digunakan dalam penelitian ini adalah Gitlab. Setelah kode berhasil ditarik ke SonarQube, SonarQube akan melakukan proses SAST dengan melakukan pemindaian kode aplikasi secara komprehensif untuk mengidentifikasi potensi kerentanan keamanan, kelemahan dalam kualitas kode, dan isu-isu lainnya tanpa perlu mengeksekusi aplikasi.



Gambar 6. Hasil SAST Menggunakan SonarQube

Hasil pengujian Static Application Security Testing (SAST) menggunakan SonarQube terhadap tiga aplikasi, yakni Juice Shop, DVJA, dan DVWA, berhasil mengidentifikasi sejumlah temuan terkait aspek keamanan, keandalan, keterpeliharaan, duplikasi kode, dan titik rawan keamanan. Hasil temuan pada aplikasi Juice Shop tercantum pada gambar 6 dan hasil dari pemindaian keseluruhan aplikasi dipaparkan pada tabel 6.

Tabel 6. Hasil SAST Menggunakan SonarQube

Aplikasi	Aspek				
	Security	Reliability	Maintability	Duplications	Security Hotspot
Juice Shop	1	46	305	8,7%	194
DVJA	0	38	36	1,1%	7
DVWA	3	63	978	10%	63

Analisis menunjukkan bahwa DVWA secara konsisten menampilkan kualitas kode paling rendah di hampir semua metrik, dengan jumlah kerentanan keamanan tertinggi (3), isu keandalan terbanyak (63), skor keterpeliharaan terburuk (978 isu), dan persentase duplikasi kode tertinggi (10%). Juice Shop, meskipun memiliki jumlah kerentanan langsung yang lebih rendah (1), menunjukkan jumlah security hotspot yang sangat signifikan (194), mengindikasikan area kode yang luas memerlukan peninjauan keamanan manual, serta tingkat keterpeliharaan yang cukup tinggi (305 isu). Sebaliknya, DVJA menampilkan metrik yang relatif lebih baik, dengan tidak adanya kerentanan langsung yang terdeteksi, keterpeliharaan yang baik (36 isu), duplikasi kode minimal (1,1%), dan jumlah security hotspot terendah (7). Angka hotspot yang tinggi mengindikasikan area kode yang memerlukan investigasi manual, menyiratkan kompleksitas atau jenis kerentanan yang mungkin berbeda.

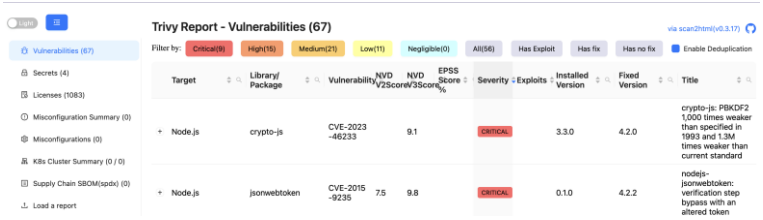
4.3 Hasil Pengujian pada Tahap Build

Setelah analisis keamanan pada level kode sumber melalui SAST, proses DevSecOps dilanjutkan ke tahap Build. Aplikasi yang dikembangkan dalam penelitian ini dijalankan di atas platform Docker dengan memanfaatkan teknologi kontainerisasi, maka pengujian keamanan pada tahap ini difokuskan pada pemindaian kontainer (container scanning). Untuk aktivitas ini, tool Trivy digunakan dengan target pemindaian image kontainer yang telah dibangun dan dijalankan.

Trivy melakukan pemindaian image kontainer untuk mengidentifikasi potensi celah keamanan yang terdapat didalam kontainer aplikasi. Hasil pemindaian ini diklasifikasikan ke dalam tiga kategori utama temuan yaitu:

- Kerentanan Aplikasi: Trivy mendeteksi kerentanan aplikasi yang berjalan dalam kontainer
- Secret Kontainer: Pemindaian terhadap secret (seperti kata sandi, kunci API, atau token) yang disimpan.
- Lisensi Aplikasi: Pemindaian lisensi dari berbagai komponen aplikasi yang terdapat di dalam kontainer.

p-ISSN: 2775-4227
e-ISSN: 2775-4219



Gambar 7. Hasil Container Scanning Menggunakan Trivy

Hasil pemindaian container menggunakan Trivy terhadap tiga aplikasi, yakni Juice Shop, DVJA, dan DVWA, berhasil mengidentifikasi sejumlah temuan terkait aspek kerentanan, pengamanan kode rahasia, dan lisensi. Hasil temuan pada aplikasi Juice Shop tercantum pada gambar 7 dan hasil dari pemindaian keseluruhan aplikasi dipaparkan pada tabel 7.

Tabel 7. Hasil Container Scanning Menggunakan Trivy

Aplikasi	Aspek		
	Kerentanan	Pengamanan Rahasia	Lisensi
Juice Shop	67 (9 Kritikal, 15 Tinggi, 21 Sedang, 11 Rendah)	4	1083
DVJA	981 (46 Kritikal, 130 Tinggi, 125 Sedang, 112 Rendah)	0	1134
DVWA	993 (1 Kritikal, 116 Tinggi, 430 Sedang, 172 Rendah)	0	1260

Untuk setiap temuan, Trivy menyediakan detail yang jelas mengenai sifat kerentanan, pengamanan rahasia yang tidak aman, atau permasalahan lisensi. Aplikasi Juice Shop menunjukkan total 67 temuan kerentanan yang terdiri dari 9 bersifat kritikal, 15 tinggi, 21 sedang, dan 11 rendah. Selain itu, terdapat 4 temuan terkait pengamanan rahasia dan 1083 isu lisensi. Sementara itu, DVJA memiliki 981 kerentanan yang meliputi 46 kritikal, 130 tinggi, 125 sedang, dan 112 rendah, tanpa temuan pengamanan rahasia, namun dengan jumlah isu lisensi sebanyak 1134. Di sisi lain, DVWA menunjukkan jumlah kerentanan tertinggi yaitu 993, yang terdiri dari 1 kritikal, 116 tinggi, 430 sedang, dan 172 rendah, juga tanpa temuan pada aspek pengamanan rahasia, namun dengan jumlah isu lisensi tertinggi yakni 1260.

Dari hasil tersebut dapat disimpulkan bahwa DVJA dan DVWA memiliki tingkat kerentanan dan isu lisensi yang lebih signifikan dibandingkan Juice Shop. Trivy menyertakan rekomendasi perbaikan atau langkah-langkah mitigasi yang dapat diambil. Informasi komprehensif ini akan membantu tim pengembang untuk segera melakukan tindakan mitigasi dan perbaikan pada aplikasi yang berjalan pada kontainer, seperti memperbarui paket yang rentan, mengelola secret dengan lebih aman, dan memastikan kepatuhan lisensi, sehingga menghasilkan aplikasi yang lebih aman sebelum di-deploy ke lingkungan selanjutnya.

4.4 Hasil Pengujian pada Tahap Test

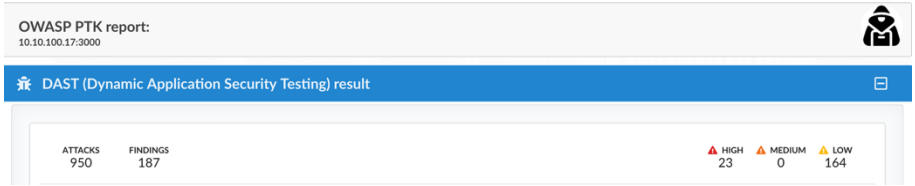
Setelah melewati tahap Build yang bertujuan memastikan keamanan container melalui pemindaian statis, siklus DevSecOps kemudian berlanjut ke tahap Test. Pada tahap ini, fokus pengujian beralih dari pemeriksaan statis ke evaluasi keamanan secara dinamis terhadap infrastruktur dan aplikasi yang sedang berjalan. Untuk mendukung proses pengujian dinamis ini, penelitian memanfaatkan metode Dynamic Application Security Testing (DAST) dengan menggunakan OWASP Penetration Testing Kit (PTK).

OWASP Penetration Testing Kit (PTK) merupakan seperangkat alat yang dikembangkan oleh OWASP (Open Web Application Security Project) yang digunakan untuk melakukan pengujian keamanan aplikasi secara dinamis. OWASP PTK membantu mendeteksi kelemahan keamanan yang muncul saat aplikasi dijalankan, seperti kesalahan konfigurasi, kelemahan dalam autentikasi, injeksi, atau kebocoran informasi sensitif.

DAST (Dynamic Application Security Testing) sendiri adalah metode pengujian keamanan aplikasi yang dilakukan saat aplikasi sedang berjalan (runtime). Dalam penelitian ini, DAST dilakukan dengan cara mengakses dan membuka seluruh tautan aplikasi yang tersedia dalam masing-masing dari ketiga aplikasi, yaitu Juice Shop, DVJA, dan DVWA. Selama proses ini, OWASP PTK merekam setiap interaksi dan mengidentifikasi potensi kerentanan berdasarkan respons yang diberikan oleh aplikasi.

Hasil dari proses DAST ini dituangkan dalam sebuah laporan yang bersifat menyeluruh, di mana setiap kerentanan yang terdeteksi akan dicatat lengkap beserta jumlah total temuan dan klasifikasinya berdasarkan tingkat keparahan, yaitu High (tinggi), Medium (sedang), dan Low (rendah).

p-ISSN: 2775-4227
e-ISSN: 2775-4219



Gambar 8. Hasil DAST Menggunakan OWASP PTK

Hasil pengujian Dynamic Application Security Testing (DAST) menggunakan OWASP PTK terhadap tiga aplikasi, yakni Juice Shop, DVJA, dan DVWA, berhasil mengidentifikasi sejumlah temuan terkait kerentanan aplikasi. Hasil temuan pada aplikasi Juice Shop tercantum pada gambar 8 dan hasil dari pemindaian keseluruhan aplikasi dipaparkan pada tabel 8.

Tabel 8. Hasil DAST Menggunakan OWASP PTK

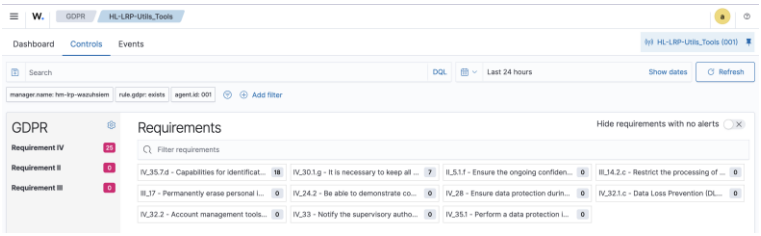
Aplikasi	Kerentanan		
	Tinggi	Sedang	Rendah
Juice Shop	23	0	164
DVJA	15	0	76
DVWA	7	0	137

Berdasarkan hasil pengujian Dynamic Application Security Testing (DAST) menggunakan OWASP Penetration Testing Kit (PTK), dapat disimpulkan bahwa ketiga aplikasi, yaitu Juice Shop, DVJA, dan DVWA, masih mengandung sejumlah kerentanan pada saat dijalankan Aplikasi Juice Shop menunjukkan jumlah kerentanan tertinggi secara keseluruhan, dengan 23 kerentanan pada tingkat tinggi dan 164 pada tingkat rendah, serta tidak ditemukan kerentanan pada tingkat sedang. Selanjutnya, aplikasi DVJA teridentifikasi memiliki 15 kerentanan tingkat tinggi dan 76 kerentanan tingkat rendah, juga tanpa adanya kerentanan tingkat sedang. Sementara itu, aplikasi DVWA mencatatkan jumlah kerentanan tingkat tinggi yang paling sedikit, yaitu sebanyak 7, dengan 137 kerentanan tingkat rendah, dan tidak ditemukan kerentanan tingkat sedang.

4.5 Hasil Pengujian pada Tahap Deploy

Setelah melalui serangkaian pengujian keamanan pada tahap sebelumnya, siklus DevSecOps berlanjut ke tahap Deploy. Sebelum aplikasi secara resmi diluncurkan atau perubahan diterapkan ke lingkungan produksi, langkah krusial yang dilakukan adalah tinjauan keamanan (security review) dengan fokus utama pada pemenuhan standar kepatuhan (compliance). Proses ini bertujuan untuk memastikan bahwa aplikasi dan infrastruktur pendukungnya telah selaras dengan regulasi dan kebijakan keamanan yang relevan.

Untuk mendukung proses tinjauan kepatuhan ini, penelitian menggunakan modul Compliance Monitoring yang terdapat pada platform Wazuh. Modul ini memungkinkan Wazuh untuk melakukan pemantauan kepatuhan terhadap berbagai target, meliputi server, kontainer, dan aplikasi yang terinstal. Wazuh bekerja dengan cara memeriksa konfigurasi dan status sistem terhadap serangkaian aturan dan standar kepatuhan yang telah ditentukan sebelumnya atau dikustomisasi. Hasil dari pemantauan kepatuhan ini kemudian disajikan oleh Wazuh, yang menunjukkan sejauh mana sistem atau aplikasi telah memenuhi persyaratan kepatuhan yang ditetapkan.



Gambar 9. Hasil Compliance Assessment Menggunakan Wazuh Compliance

Penelitian ini menggunakan regulasi GDPR sebagai kriteria kepatuhan dan Wazuh menghasilkan pemindaian sesuai yang tertera pada Gambar 8. Laporan Wazuh tersebut mengindikasikan sejauh mana aplikasi dan infrastruktur pendukungnya telah memenuhi berbagai persyaratan GDPR. Pemenuhan kebutuhan GDPR

p-ISSN: 2775-4227
e-ISSN: 2775-4219

menandakan bahwa sistem telah mengimplementasikan langkah-langkah teknis dan organisasional yang sesuai untuk melindungi data pribadi sebagaimana diamanatkan oleh regulasi tersebut.

Secara spesifik, teridentifikasi empat requirement yang dipenuhi, ditandai dengan tingginya jumlah event tercatat yang menunjukkan pemantauan aktif:

- Requirement II_5.1.f: Menjamin kerahasiaan, integritas, ketersediaan, dan ketahanan sistem pemrosesan secara berkelanjutan.
- Requirement IV_35.7.d: Menyediakan kapabilitas identifikasi, pemblokiran, dan investigasi forensik terhadap pelanggaran data, didukung oleh perangkat keamanan dan analisis perilaku.
- Requirement IV_30.1.g: Mengharuskan pendokumentasian seluruh aktivitas pemrosesan dan pemeliharaan inventaris data.
- Requirement IV_32.2: Mengamanatkan penggunaan perangkat manajemen akun untuk pemantauan ketat dan kontrol akses data.

Frekuensi event yang masif untuk kebutuhan yang terpenuhi ini mengindikasikan bahwa Wazuh secara terus menerus mencatatkan aktivitas / kejadian yang relevan, sejalan dengan prinsip akuntabilitas dan pencatatan dalam regulasi perlindungan data terutama sejalan dengan prinsip pencatatan pada regulasi perlindungan data seperti UU PDP Pasal 31. Sebaliknya, adanya kebutuhan dengan jumlah pemenuhan nol mengisyaratkan bahwa aplikasi dan infrastruktur terkait belum berhasil memenuhi aspek-aspek GDPR tersebut.

4.6 Hasil Pengujian pada Tahap Operate

Setelah aplikasi berhasil diluncurkan ke lingkungan produksi pasca verifikasi pada tahap Deploy, siklus DevSecOps memasuki fase operasional berkelanjutan, yaitu tahap Operate. Dalam tahap Operate, penelitian memanfaatkan kapabilitas Vulnerability Assessment (VA) yang terdapat pada aplikasi Nessus Vulnerability Scan. Nessus Vulnerability Scan dirancang untuk melakukan pemindaian terhadap berbagai target, termasuk server fisik maupun virtual, kontainer yang sedang berjalan, serta aplikasi yang terpasang dan aktif di dalamnya. Nessus mengumpulkan informasi detail mengenai konfigurasi dan versi perangkat lunak dari target-target tersebut. Informasi ini kemudian dibandingkan dan dicocokkan dengan basis data kerentanan Nessus yang terus diperbarui. Basis data ini berisi daftar kerentanan yang telah diketahui publik (seperti CVEs) beserta tingkat kritikalitasnya.

Gambar 10. Hasil Vulnerability Assessment (VA) Menggunakan Nessus Vulnerability Scan

Hasil pengujian Vulnerability Assessment (VA) menggunakan Nessus Vulnerability Scan terhadap tiga aplikasi, yakni Juice Shop, DVJA, dan DVWA, berhasil mengidentifikasi sejumlah temuan terkait kerentanan aplikasi. Hasil temuan pada aplikasi Juice Shop tercantum pada gambar 10 dan hasil dari pemindaian keseluruhan aplikasi dipaparkan pada tabel 9.

Tabel 9. Hasil Vulnerability Assessment (VA) Menggunakan Nessus Vulnerability Scan

Aplikasi	Kerentanan			
	Kritis	Tinggi	Sedang	Rendah
Juice Shop	0	0	2	1
DVJA	17	43	43	6
DVWA	0	3	7	1

Analisis hasil Vulnerability Assessment menunjukkan perbedaan signifikan dalam profil keamanan ketiga aplikasi tersebut. Aplikasi DVJA tercatat memiliki tingkat kerentanan paling substansial, dengan terdeteksinya 17 kerentanan berkategori 'Kritis', 43 kerentanan 'Tinggi', 43 'Sedang', dan 6 'Rendah'. Jumlah temuan yang sangat

p-ISSN: 2775-4227
e-ISSN: 2775-4219

tinggi pada kategori 'Kritis' dan 'Tinggi' pada DVJA mengindikasikan adanya celah keamanan serius yang dapat dieksploitasi oleh pihak tidak bertanggung jawab, berpotensi menyebabkan dampak kerusakan yang signifikan atau pengambilalihan sistem. Aplikasi DVWA menunjukkan 3 kerentanan dengan tingkat risiko 'Tinggi', 7 'Sedang', dan 1 'Rendah'. Sementara itu, aplikasi Juice Shop memperlihatkan profil risiko yang relatif lebih rendah dibandingkan dua aplikasi lainnya, dengan temuan 2 kerentanan 'Sedang' dan 1 'Rendah', tanpa adanya kerentanan yang tergolong 'Kritis' maupun 'Tinggi'.

4.7 Hasil Pengujian pada Tahap Monitor

Setelah aplikasi berhasil diluncurkan ke lingkungan produksi pasca verifikasi pada tahap Deploy, siklus DevSecOps memasuki fase operasional berkelanjutan, yaitu tahap Monitor. Tahap ini krusial untuk memastikan keamanan aplikasi dan infrastruktur secara real-time selama masa penggunaan aktif. Dua aktivitas utama yang menjadi fokus dalam tahap ini adalah implementasi Security Information and Event Management (SIEM) dan File Integrity Monitoring (FIM).

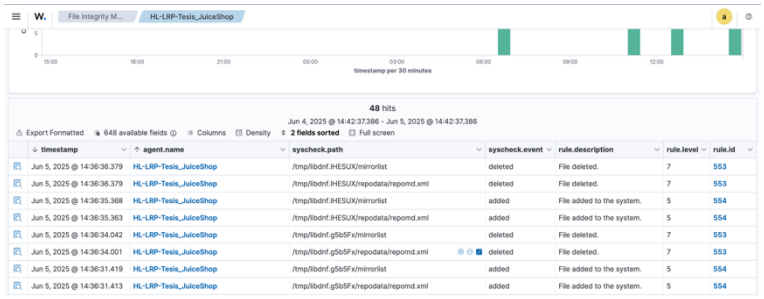
Aktivitas SIEM berperan dalam operasional dengan melakukan pemantauan dalam aspek proses dan keamanan aplikasi dengan cara mengumpulkan dan menganalisis log serta data peristiwa (event) dari berbagai sumber dalam infrastruktur TI, termasuk server, jaringan, dan aplikasi itu sendiri. Dengan adanya SIEM, tim keamanan dapat memantau secara terus-menerus aktivitas yang terjadi saat aplikasi berjalan, mendeteksi anomali atau pola yang mencurigakan, serta merespons insiden keamanan secara lebih cepat dan efektif.

Implementasi SIEM pada aplikasi menghasilkan peristiwa (event) yang secara lengkap direkam dalam bentuk log peristiwa. Log peristiwa seperti gambar 11 menghasilkan data lengkap seperti proses yang sedang berjalan, port jaringan yang digunakan, lalu lintas data, dan peringatan jika aktivitas yang dilakukan menjadi potensi ancaman oleh Wazuh.

Gambar 11. Hasil Implementasi Monitoring Kejadian dengan Wazuh SIEM

Selanjutnya, aktivitas File Integrity Monitoring (FIM) bertujuan untuk menjaga integritas sistem dan data dengan cara memantau perubahan pada berkas-berkas dan direktori penting. FIM secara aktif melacak setiap berkas yang dibuat, diubah, atau dihapus pada sistem yang dipantau selama proses operasional dan pemantauan berlangsung. Jika terjadi perubahan yang tidak sah atau mencurigakan pada berkas konfigurasi kritis, berkas sistem, atau data aplikasi, FIM akan memberikan peringatan.

Hal ini memungkinkan deteksi dini terhadap potensi intrusi, infeksi malware, atau perubahan yang tidak diotorisasi, sehingga tindakan perbaikan dapat segera dilakukan untuk menjaga stabilitas dan keamanan sistem.



Gambar 12. Hasil Implementasi Monitoring Integritas Data dengan Wazuh FIM

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Implementasi FIM pada aplikasi menghasilkan kejadian (event) yang secara lengkap direkam dalam bentuk log kejadian. Log peristiwa seperti gambar 12 menghasilkan data lengkap seperti data yang ditambahkan, dikurangi, atau dihapus, serta peringatan jika ada perubahan data yang tidak diinginkan menjadi potensi ancaman oleh Wazuh.

4.8 Diskusi Hasil Tahap Code

Tabel 10. Perbandingan Hasil SAST terhadap Penelitian Sebelumnya

Aplikasi	Tingkat Kerentanan		
	Tinggi	Sedang	Rendah
DVJA SonarQube	1	2	4
SAST DVJA oleh Aljohani et al [15]	0	1	3
DVWA SonarQube	7	12	47
SAST DVWA oleh Marandi et al [16]	29	15	1

Perbedaan ini dapat dibandingkan dengan penelitian sebelumnya. Penelitian oleh Aljohani et al. [15] dan Marandi et al. [16] cenderung lebih terfokus pada identifikasi dan klasifikasi kerentanan keamanan saja berdasarkan tingkat keparahannya. Sebaliknya, penelitian ini memanfaatkan perangkat seperti SonarQube, yang tidak hanya mendeteksi kerentanan keamanan tetapi juga mampu menganalisis berbagai aspek kualitas kode lainnya, termasuk keandalan, keterpeliharaan, duplikasi kode, dan security hotspots.

Seperti yang ditunjukkan pada Tabel 10, penelitian ini berhasil mengidentifikasi temuan yang secara kuantitatif lebih banyak dan memiliki cakupan aspek yang lebih luas dibandingkan dengan penelitian sebelumnya. Secara kuantitatif berdasarkan jumlah total kerentanan yang dikategorikan berdasarkan tingkat keparahan (Tinggi, Sedang, Rendah), penelitian ini memang menunjukkan jumlah total temuan yang lebih banyak untuk kedua aplikasi (7 vs 4 untuk DVJA; 66 vs 45 untuk DVWA) dibandingkan dengan penelitian sebelumnya yang direferensikan.

Temuan ini mengonfirmasi efektivitas SonarQube SAST dalam mendeteksi kelemahan pada aplikasi dengan desain tidak aman secara eksplisit. Temuan ini menyediakan masukan kuantitatif dan kualitatif yang krusial bagi tim pengembang sebagai dasar untuk melakukan perbaikan dan penguatan kode aplikasi sebelum dilanjutkan ke tahapan berikutnya dalam siklus DevSecOps.

5. KESIMPULAN

Berdasarkan serangkaian pengujian yang telah dilaksanakan pada setiap tahapan siklus DevSecOps, mulai dari Perencanaan (Plan) hingga Operasional dan Pemantauan (Operate and Monitor), dapat ditarik kesimpulan bahwa pendekatan yang diterapkan berhasil mengidentifikasi celah keamanan secara komprehensif. Setiap fase, dengan aktivitas spesifik dan perangkat (tools) yang telah dipilih dan diimplementasikan, secara kolektif memberikan gambaran menyeluruh mengenai potensi risiko dan kerentanan yang ada pada aplikasi maupun infrastruktur pendukungnya.

Dibandingkan dengan penelitian sebelumnya di bidang DevSecOps, studi ini menunjukkan keunggulan:

- Penelitian ini berhasil mengimplementasikan siklus DevSecOps yang lebih lengkap, mencakup seluruh alur pengembangan dan operasional.
- Aktivitas yang dirancang untuk setiap tahapan terbukti relevan dan mampu mengevaluasi aspek keamanan yang kritikal pada masing-masing fase.
- Pemilihan tools dilakukan secara cermat berdasarkan kriteria yang ditetapkan, memastikan kesesuaian fungsional dan konteks implementasi.

Penelitian ini memastikan bahwa seluruh proses dalam siklus DevSecOps tercakup dan setiap aktivitas keamanan yang esensial dilengkapi dengan setidaknya satu perangkat pendukung. Hal ini berbeda dengan beberapa studi sebelumnya yang mungkin hanya fokus pada sebagian aktivitas atau tidak mengintegrasikan perangkat secara menyeluruh pada setiap tahapan.

Dengan demikian, temuan celah keamanan yang dihasilkan dari implementasi ini menjadi sangat penting dan bernilai tinggi. Informasi ini berfungsi sebagai umpan balik (feedback) yang komprehensif dan terstruktur bagi tim pengembang (Dev), tim keamanan (Sec), dan tim operasional (Ops). Adanya feedback loop yang jelas ini krusial untuk memfasilitasi kolaborasi antar tim, memungkinkan perbaikan berkelanjutan, dan pada akhirnya meningkatkan postur keamanan aplikasi secara keseluruhan.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

DAFTAR PUSTAKA

[1] R. Mao *et al.*, "Preliminary Findings about DevSecOps from Grey Literature," in *Proceedings - 2020 IEEE 20th International Conference on Software Quality, Reliability, and Security, QRS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 450–457. doi: 10.1109/QRS51102.2020.00064.

[2] T. Rangnau, R. V. Buijtenen, F. Fransen, and F. Turkmen, "Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines," in *Proceedings - 2020 IEEE 24th International Enterprise Distributed Object Computing Conference, EDOC 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 145–154. doi: 10.1109/EDOC49727.2020.00026.

[3] K. Byrne and A. Cevenini, "Aligning DevOps Concepts with Agile Models of the Software Development Life Cycle (SLDC) in Pursuit of Continuous Regulatory Compliance," 2023, pp. 359–374. doi: 10.1007/978-3-031-29078-7_32.

[4] Surfshark, "Data Breach Statistics & Trends: Global & by Country," May 2024. [Online]. Available: https://surfshark.com/research/data-breach-monitoring?country=id

[5] M. Chen, B. Liang, and X. Lu, "The Practice and Application of a Novel DevSecOps Platform on Security," in *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology, AINIT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 558–562. doi: 10.1109/AINIT61980.2024.10581700.

[6] A. Caniglia, V. Dentamaro, S. Galantucci, and D. Impedovo, "FOBICS: Assessing project security level through a metrics framework that evaluates DevSecOps performance," *Inf Softw Technol*, vol. 178, Feb. 2025, doi: 10.1016/j.infsof.2024.107605.

[7] J. Immaneni, "Securing Fintech with DevSecOps: Scaling DevOps with Compliance in Mind," *Journal of Big Data and Smart Systems*, vol. 2, no. 1, pp. 1–8, 2021, [Online]. Available: https://universe-publisher.com/index.php/jbds/article/view/24

[8] IBM, "Cost of a Data Breach Report 2023," Jul. 2023. [Online]. Available: https://www.ibm.com/reports/data-breach

[9] M. Asif, Y. Javed, and M. Hussain, "Automated Analysis of Pakistani Websites' Compliance with GDPR and Pakistan Data Protection Act," in *2021 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, Dec. 2021, pp. 231–236. doi: 10.1109/FIT53090.2021.00050.

[10] S. D. Rosadi, *Undang-Undang Pelindungan Data Pribadi (UU PDP): UU RI NO. 27 Tahun 2022 Disertai Pembahasan*. Jakarta, Indonesia: Sinar Grafika, 2023.

[11] C. Feio, N. Santos, N. Escravana, and B. Pacheco, "An Empirical Study of DevSecOps Focused on Continuous Security Testing," in *Proceedings - 9th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 610–617. doi: 10.1109/EuroSPW61312.2024.00074.

[12] P. David, M. K. Kushwaha, and G. Suseela, "DevSecOps in Finance: Strengthening the Security Model of Applications," in *2023 4th IEEE International Conference on Data Engineering and Communication Systems (ICDECS)*, Bangalore, India, Aug. 2023, pp. 1–6. doi: 10.1109/ICDECS59460.2023.10353518.

[13] J. Martelleur and A. Hamza, "Security Tools in DevSecOps: A Systematic Literature Review," Karlskrona, Sweden, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:bth-23260

[14] W.-T. Lee and Z.-W. Liu, "Microservices-based DevSecOps Platform using Pipeline and Open Source Software," *Journal of Information Science and Engineering*, vol. 39, no. 5, pp. 1117–1128, Sep. 2023, doi: 10.6688/JISE.202309_39(5).0007.

[15] M. A. Aljohani and S. S. Alqahtani, "A Unified Framework for Automating Software Security Analysis in DevSecOps," in *International Conference on Smart Computing and Application, ICSCA 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICSCA57840.2023.10087568.

[16] M. Marandi, A. Bertia, and S. Silas, "Implementing and Automating Security Scanning to a DevSecOps CI/CD Pipeline," in *2023 World Conference on Communication and Computing, WCONF 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/WCONF58270.2023.10235015.