Jurnal Riset Multidisiplin Edukasi
https://journal.hasbaedukasi.co.id/index.php/jurmie
Halaman : 1157 - 1172

Analisis Penerapan *Secure Software Development Lifecycle* (SSDLC) dalam Meningkatkan Keamanan Aplikasi Berbasis *Cloud*

Septaro Travian Gadha[1], Dina Zatusiva Haq[2]
Program Studi Informatika, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur [1,2]
Email: 23081010270@student.upnjatim.ac.id[1], dinaza.if@upnjatim.ac.id[2]

Informasi

Volume : 3
Nomor : 6
Bulan : Juni
Tahun : 2026
E-ISSN : 3062-9624

Abstract

Perkembangan komputasi awan menghadirkan tantangan keamanan yang kompleks, sering kali akibat kurangnya integrasi keamanan dalam siklus pengembangan perangkat lunak. Penelitian ini bertujuan menganalisis penerapan Secure Development Lifecycle (SDL) untuk meningkatkan keamanan aplikasi berbasis cloud. Melalui metode studi literatur deskriptif kualitatif, hasil kajian menunjukkan bahwa integrasi praktik SDL secara menyeluruh—mulai dari perencanaan hingga monitoring—efektif mengidentifikasi dan memitigasi kerentanan sejak tahap awal. Pendekatan sistematis dan proaktif ini terbukti relevan dalam menekan risiko kerentanan umum seperti kelemahan kontrol akses, miskonfigurasi, dan desain yang tidak aman, sehingga menjamin keamanan aplikasi cloud secara berkelanjutan.

Keyword: *Secure Development Lifecycle, keamanan aplikasi, cloud computing, aplikasi berbasis cloud, keamanan perangkat lunak.*

A. PENDAHULUAN

Perkembangan teknologi komputasi awan telah mendorong perubahan signifikan dalam cara aplikasi dirancang, dikembangkan, dan dioperasikan. Lingkungan cloud menawarkan keunggulan berupa skalabilitas, fleksibilitas, dan efisiensi operasional, sehingga banyak aplikasi modern dibangun dengan memanfaatkan layanan cloud sebagai fondasi utama pengelolaan data dan proses bisnis. Meskipun demikian, peningkatan adopsi cloud juga diikuti oleh bertambahnya kompleksitas keamanan, terutama karena aplikasi berbasis cloud umumnya berjalan pada lingkungan yang dinamis, terdistribusi, dan terhubung dengan berbagai layanan, komponen, serta dependensi pihak ketiga.

Tantangan keamanan pada aplikasi berbasis cloud tidak hanya berkaitan dengan infrastruktur, tetapi juga sangat dipengaruhi oleh proses pengembangan perangkat lunak itu sendiri. Berbagai kerentanan pada aplikasi modern, seperti broken access control, security misconfiguration, dan insecure design, menunjukkan bahwa banyak risiko keamanan muncul sejak tahap perancangan dan implementasi, bukan hanya pada tahap operasional. Kondisi tersebut menunjukkan bahwa pendekatan keamanan yang hanya dilakukan pada tahap akhir

pengembangan tidak lagi memadai, terutama pada lingkungan cloud yang menuntut kecepatan rilis, perubahan konfigurasi yang terus-menerus, serta integrasi berkelanjutan antara kode, infrastruktur, dan layanan (Satrinia Dwina, 2022).

Sebagai respons terhadap kebutuhan tersebut, Secure Development Lifecycle (SDL) menjadi salah satu pendekatan yang relevan untuk diterapkan dalam pengembangan aplikasi berbasis cloud. Microsoft menempatkan praktik seperti penetapan standar keamanan, penggunaan komponen yang terbukti aman, security design review, threat modeling, pengujian keamanan, keamanan platform operasional, serta monitoring dan response sebagai bagian penting dari SDL. Di sisi lain, prinsip secure development dari NCSC menekankan bahwa layanan cloud harus dirancang, dikembangkan, dan di-deploy dengan cara yang mampu meminimalkan serta memitigasi ancaman keamanan. Dengan demikian, SDL tidak hanya berfungsi sebagai prosedur teknis, tetapi juga sebagai kerangka kerja sistematis untuk mengintegrasikan keamanan ke dalam seluruh siklus hidup pengembangan perangkat lunak.

Penerapan SDL pada aplikasi berbasis cloud menjadi semakin penting karena karakteristik cloud memperluas attack surface dan mempercepat siklus perubahan sistem. Penggunaan arsitektur cloud-native, container, microservices, dan pipeline pengembangan yang cepat menuntut pengamanan yang tidak berhenti pada kode program, tetapi juga mencakup desain, konfigurasi, supply chain perangkat lunak, serta pemantauan saat aplikasi telah berjalan. Jika keamanan tidak dipertimbangkan sejak awal, maka organisasi berisiko menghadapi kebocoran data, penyalahgunaan akses, kesalahan konfigurasi, dan gangguan layanan yang dapat berdampak pada kerahasiaan, integritas, dan ketersediaan sistem.

Berdasarkan kondisi tersebut, diperlukan kajian yang menganalisis bagaimana Secure Development Lifecycle dapat diterapkan untuk meningkatkan keamanan aplikasi berbasis cloud. Penelitian ini menggunakan pendekatan studi literatur untuk menelaah konsep, tahapan, dan praktik SDL dari berbagai sumber ilmiah dan referensi standar keamanan perangkat lunak. Melalui pendekatan tersebut, penelitian ini diharapkan dapat memberikan pemahaman yang lebih sistematis mengenai kontribusi setiap tahapan SDL terhadap penguatan keamanan aplikasi cloud, sekaligus mengidentifikasi tantangan penerapannya dalam lingkungan pengembangan modern.

Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana konsep dan tahapan Secure Development Lifecycle diterapkan dalam pengembangan aplikasi berbasis cloud?
2. Bagaimana kontribusi setiap tahapan Secure Development Lifecycle dalam meningkatkan keamanan aplikasi berbasis cloud terhadap ancaman dan kerentanan yang umum terjadi?
3. Apa saja tantangan yang dihadapi dalam penerapan Secure Development Lifecycle pada pengembangan aplikasi berbasis cloud?
4. Bagaimana rekomendasi penerapan Secure Development Lifecycle yang dapat mendukung peningkatan keamanan aplikasi berbasis cloud secara lebih sistematis dan berkelanjutan?

Tujuan Penelitian

Sejalan dengan rumusan masalah tersebut, tujuan penelitian ini adalah:

1. Menganalisis konsep dan tahapan Secure Development Lifecycle dalam pengembangan aplikasi berbasis cloud.
2. Mengidentifikasi kontribusi setiap tahapan Secure Development Lifecycle terhadap peningkatan keamanan aplikasi berbasis cloud.
3. Mengidentifikasi tantangan yang memengaruhi penerapan Secure Development Lifecycle pada pengembangan aplikasi berbasis cloud.
4. Menyusun rekomendasi konseptual terkait penerapan Secure Development Lifecycle untuk mendukung keamanan aplikasi berbasis cloud.

Manfaat Penelitian

Manfaat penelitian ini dapat dilihat dari dua sisi, yaitu manfaat teoritis dan manfaat praktis.

Manfaat teoritis

Penelitian ini diharapkan dapat memperkaya kajian akademik di bidang keamanan perangkat lunak, khususnya mengenai penerapan Secure Development Lifecycle pada pengembangan aplikasi berbasis cloud. Selain itu, penelitian ini dapat menjadi referensi konseptual bagi penelitian lanjutan yang membahas integrasi keamanan ke dalam siklus hidup pengembangan perangkat lunak modern.

Manfaat praktis

Secara praktis, penelitian ini diharapkan dapat memberikan gambaran yang sistematis mengenai tahapan dan praktik Secure Development Lifecycle yang relevan untuk meningkatkan keamanan aplikasi berbasis cloud. Hasil penelitian ini juga dapat menjadi bahan

pertimbangan bagi pengembang, mahasiswa, maupun pihak yang mempelajari keamanan aplikasi dalam memahami pentingnya integrasi aspek keamanan sejak tahap perencanaan hingga operasional aplikasi.

Tinjauan Pustaka

1. Cloud Computing dan Aplikasi Berbasis Cloud

Cloud computing merupakan model komputasi yang memungkinkan akses jaringan secara luas, nyaman, dan sesuai kebutuhan terhadap kumpulan sumber daya komputasi terkonfigurasi bersama, seperti server, penyimpanan, aplikasi, dan layanan, yang dapat disediakan dan dilepas dengan cepat dengan upaya manajemen minimal. Definisi ini menunjukkan bahwa cloud computing tidak sekadar tempat penyimpanan data, tetapi merupakan lingkungan komputasi yang mendukung penyediaan layanan teknologi secara fleksibel dan elastis.

Dalam perkembangannya, cloud computing menjadi fondasi penting bagi pengembangan aplikasi modern karena menawarkan skalabilitas, efisiensi biaya, dan fleksibilitas operasional. Aplikasi berbasis cloud memanfaatkan infrastruktur cloud untuk menjalankan proses bisnis, pengelolaan data, serta integrasi layanan digital secara lebih dinamis dibandingkan aplikasi tradisional yang berjalan pada lingkungan lokal. Karakteristik ini memungkinkan organisasi mengembangkan dan mendistribusikan aplikasi dengan lebih cepat, tetapi juga menimbulkan tantangan baru dalam aspek kontrol, konfigurasi, dan pengamanan layanan (Asrorwadi et al., 2014).

Selain itu, aplikasi berbasis cloud umumnya dibangun dengan arsitektur yang semakin kompleks, seperti pemanfaatan API, layanan pihak ketiga, microservices, serta mekanisme deployment yang otomatis dan berkelanjutan. Kompleksitas tersebut memperluas ruang paparan risiko karena keamanan aplikasi tidak lagi hanya ditentukan oleh kualitas kode program, tetapi juga oleh bagaimana layanan cloud dikonfigurasi, diintegrasikan, dan dioperasikan dalam lingkungan yang terus berubah.

2. Keamanan Aplikasi Berbasis Cloud

Keamanan aplikasi berbasis cloud mengacu pada upaya perlindungan terhadap aplikasi, data, dan layanan yang berjalan di lingkungan cloud dari ancaman seperti akses tidak sah, kebocoran data, manipulasi layanan, serta penyalahgunaan sumber daya. Dalam konteks ini, keamanan tidak hanya berfokus pada pencegahan serangan dari luar, tetapi juga pada pengendalian risiko yang muncul dari konfigurasi yang salah, komponen pihak ketiga yang rentan, serta kelemahan desain aplikasi (Sugiantoro, 2025).

Lingkungan cloud memiliki sifat dinamis, multi-tenant, dan terdistribusi, sehingga model pengamanan yang bersifat tradisional dan reaktif menjadi semakin tidak memadai. Aplikasi modern yang dibangun di atas cloud sering kali mengalami perubahan cepat melalui pipeline CI/CD, integrasi layanan eksternal, dan deployment berulang, yang berarti setiap kesalahan desain atau implementasi dapat dengan cepat menyebar ke lingkungan produksi. Oleh karena itu, keamanan aplikasi cloud menuntut pendekatan yang menyeluruh dan berkelanjutan dari tahap perencanaan hingga tahap operasional.

Keamanan aplikasi berbasis cloud juga tidak dapat dipisahkan dari prinsip perlindungan kerahasiaan, integritas, dan ketersediaan data serta layanan. Jika aplikasi dikembangkan tanpa mempertimbangkan kontrol keamanan yang memadai, risiko seperti pengambilalihan akun, kesalahan autentikasi, paparan data sensitif, dan gangguan layanan dapat muncul dan berdampak langsung pada kualitas layanan maupun kepercayaan pengguna.

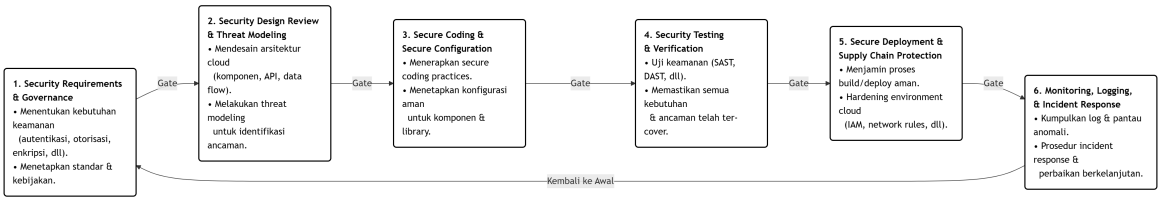
3. Secure Development Lifecycle

Secure Development Lifecycle (SDL) merupakan pendekatan pengembangan perangkat lunak yang mengintegrasikan praktik keamanan ke dalam setiap tahapan siklus hidup pengembangan perangkat lunak. Pendekatan ini bertujuan untuk memastikan bahwa aspek keamanan tidak ditempatkan sebagai aktivitas tambahan di akhir proses, melainkan menjadi bagian yang melekat sejak tahap perencanaan, perancangan, implementasi, pengujian, deployment, hingga pemeliharaan aplikasi (Akmala et al., 2021).

Microsoft menjelaskan SDL sebagai sekumpulan praktik utama yang membantu organisasi mengintegrasikan keamanan ke dalam keseluruhan proses pengembangan. Praktik tersebut mencakup penetapan standar, metrik, dan tata kelola keamanan; penggunaan fitur, bahasa, dan framework yang telah terbukti aman; security design review dan threat modeling; standar kriptografi; pengamanan software supply chain; pengamanan engineering environment; security testing; operational platform security; serta security monitoring and response. Rangkaian praktik ini menunjukkan bahwa SDL bersifat sistematis dan dirancang untuk mengurangi kerentanan sejak tahap awal pengembangan, bukan hanya mendeteksi masalah setelah aplikasi selesai dibuat.

Dalam konteks aplikasi berbasis cloud, SDL menjadi semakin relevan karena lingkungan cloud menuntut integrasi keamanan yang lebih luas, meliputi kode, konfigurasi, komponen pihak ketiga, infrastruktur, dan proses deployment. Threat modeling, secure coding, pengujian keamanan, dan monitoring berkelanjutan menjadi elemen penting untuk memastikan bahwa aplikasi cloud dibangun secara secure by design dan mampu merespons ancaman secara lebih

proaktif. Dengan demikian, SDL dapat dipandang sebagai kerangka kerja konseptual yang mendukung pembangunan aplikasi cloud yang lebih aman dan berkelanjutan.



Gambar 1 Tahapan Secure Development Lifecycle dalam Pengembangan Aplikasi Berbasis Cloud

Gambar diatas menggambarkan tahapan utama Secure Development Lifecycle dalam konteks pengembangan aplikasi berbasis cloud, yang dimulai dari penetapan kebutuhan dan tata kelola keamanan, dilanjutkan dengan security design review dan threat modeling, penerapan secure coding dan konfigurasi aman, pengujian serta verifikasi keamanan, hingga tahap secure deployment dan aktivitas monitoring, logging, serta incident response secara berkelanjutan. Setiap tahapan saling terhubung membentuk siklus yang berulang, sehingga hasil pemantauan dan penanganan insiden pada fase operasional dapat digunakan sebagai umpan balik untuk memperbaiki kebutuhan, desain, maupun implementasi pada iterasi pengembangan berikutnya. Dengan demikian, gambar ini menegaskan bahwa penerapan Secure Development Lifecycle pada aplikasi berbasis cloud bukan sekadar rangkaian langkah terpisah, tetapi sebuah proses berkesinambungan yang mengintegrasikan keamanan di seluruh siklus hidup pengembangan perangkat lunak.

4. Kerentanan dan Ancaman Keamanan Aplikasi

Pembahasan mengenai kerentanan aplikasi modern sering merujuk pada OWASP Top 10 sebagai salah satu acuan utama untuk mengidentifikasi risiko keamanan aplikasi web. Daftar OWASP Top 10:2021 mencakup Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components, Identification and Authentication Failures, Software and Data Integrity Failures, Security Logging and Monitoring Failures, serta Server-Side Request Forgery (SSRF). Daftar ini penting karena memberikan gambaran bahwa ancaman pada aplikasi modern mencakup kelemahan desain, kelemahan implementasi, serta kegagalan pengawasan dan operasional.

OWASP menempatkan Broken Access Control sebagai kategori risiko utama, sementara Insecure Design diperkenalkan sebagai kategori yang menegaskan pentingnya desain aman dan threat modeling sejak awal pengembangan. Hal tersebut memperlihatkan bahwa banyak masalah keamanan aplikasi tidak bermula dari eksploitasi teknis semata, tetapi dari keputusan

desain yang lemah, kontrol akses yang tidak konsisten, serta asumsi yang salah dalam pengelolaan aplikasi dan data.

Dalam konteks aplikasi berbasis cloud, kategori seperti Security Misconfiguration, Software and Data Integrity Failures, serta Security Logging and Monitoring Failures menjadi semakin relevan karena aplikasi cloud sangat bergantung pada konfigurasi layanan, pipeline deployment, integrasi pihak ketiga, serta pengamatan operasional secara berkelanjutan. Oleh sebab itu, kerentanan dan ancaman keamanan aplikasi cloud perlu dianalisis tidak hanya dari sisi hasil akhirnya, tetapi juga dari sisi proses pengembangan yang melahirkannya.

5. Penelitian Terdahulu

Berbagai penelitian menunjukkan bahwa keamanan cloud dan keamanan aplikasi modern semakin sering dikaji dalam hubungan dengan proses pengembangan perangkat lunak. Kajian tentang keamanan CI/CD untuk cloud computing menunjukkan bahwa pengamanan aplikasi cloud memerlukan perhatian terhadap tools, proses, serta tantangan yang muncul sepanjang lifecycle deployment, termasuk akses tidak sah dan autentikasi yang lemah. Penelitian lain juga menegaskan bahwa keamanan cloud tidak dapat dipisahkan dari integrasi kontrol keamanan ke dalam software development lifecycle, khususnya pada pendekatan DevOps dan DevSecOps (Vieri fadli et al., n.d.).

Di sisi lain, literatur mengenai keamanan aplikasi dari code to cloud menunjukkan bahwa perubahan lanskap pengembangan modern menuntut strategi keamanan yang lebih proaktif dan menyeluruh, mencakup secure coding practices, pemanfaatan cloud security services, serta penguatan application security posture secara menyeluruh. Hal ini memperkuat pandangan bahwa pengamanan aplikasi cloud memerlukan pendekatan yang tidak parsial, melainkan terintegrasi dari tahap desain hingga operasional.

Meskipun demikian, sebagian besar penelitian terdahulu masih berfokus pada aspek tertentu, seperti keamanan deployment, keamanan cloud-native workflow, DevSecOps, atau mitigasi risiko keamanan cloud secara umum. Kajian yang secara khusus menempatkan Secure Development Lifecycle sebagai kerangka konseptual utama untuk menganalisis peningkatan keamanan aplikasi berbasis cloud secara menyeluruh masih relatif terbatas. Oleh karena itu, penelitian ini menempati posisi untuk mengisi celah tersebut dengan menganalisis kontribusi tahapan-tahapan SDL terhadap penguatan keamanan aplikasi cloud secara sistematis.

6. Research Gap

Berdasarkan telaah pustaka, dapat diketahui bahwa penelitian terkait keamanan cloud umumnya menyoroti risk assessment, secure coding, DevSecOps, keamanan CI/CD, atau

keamanan cloud-native application secara parsial. Sementara itu, kajian yang secara khusus menganalisis penerapan Secure Development Lifecycle sebagai kerangka konseptual untuk meningkatkan keamanan aplikasi berbasis cloud secara menyeluruh masih belum banyak ditemukan. Oleh karena itu, penelitian ini diarahkan untuk menganalisis bagaimana tahapan dalam Secure Development Lifecycle berkontribusi terhadap peningkatan keamanan aplikasi berbasis cloud, sekaligus merumuskan rekomendasi konseptual yang dapat digunakan sebagai acuan dalam pengembangan aplikasi yang lebih aman.

B. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan metode studi literatur. Pendekatan kualitatif deskriptif digunakan karena penelitian ini berfokus pada pemahaman, penelaahan, dan interpretasi konsep Secure Development Lifecycle dalam konteks peningkatan keamanan aplikasi berbasis cloud berdasarkan berbagai sumber ilmiah dan referensi yang relevan. Melalui metode studi literatur, penelitian diarahkan untuk mengidentifikasi, mengkaji, dan mensintesis temuan-temuan dari publikasi ilmiah, dokumen standar, serta panduan praktik keamanan perangkat lunak yang berkaitan dengan keamanan aplikasi cloud dan Secure Development Lifecycle.

Jenis data yang digunakan dalam penelitian ini adalah data sekunder, yaitu data yang diperoleh dari sumber-sumber tertulis tanpa pengumpulan data lapangan secara langsung. Sumber data meliputi artikel jurnal, prosiding, dokumen standar, white paper, dan panduan resmi yang membahas cloud computing, keamanan aplikasi berbasis cloud, Secure Development Lifecycle, serta kerentanan keamanan aplikasi modern. Referensi utama dalam penelitian ini mencakup pedoman Secure Development Lifecycle dari Microsoft, prinsip secure development pada cloud dari National Cyber Security Centre (NCSC), serta referensi kerentanan aplikasi dari OWASP Top 10 sebagai acuan analisis konseptual.

Teknik pengumpulan data dilakukan melalui penelusuran literatur secara terarah pada sumber-sumber akademik dan referensi resmi yang relevan dengan topik penelitian. Proses penelusuran dilakukan dengan menggunakan kata kunci yang berkaitan dengan Secure Development Lifecycle, keamanan aplikasi berbasis cloud, secure software development, cloud security, serta OWASP Top 10. Literatur yang diperoleh kemudian diseleksi berdasarkan kesesuaian topik, keterkaitan dengan tujuan penelitian, kejelasan pembahasan konsep, serta relevansi isi terhadap fokus analisis penelitian. Pada tahap ini, penelitian tidak diarahkan sebagai systematic literature review penuh berbasis PRISMA, melainkan sebagai studi literatur

terarah dengan proses identifikasi, seleksi, dan klasifikasi sumber yang dilakukan secara sistematis agar tetap relevan dan dapat dipertanggungjawabkan.

Teknik analisis data dalam penelitian ini dilakukan secara deskriptif-analitis. Data dan informasi dari berbagai literatur yang telah dipilih terlebih dahulu diidentifikasi, diklasifikasikan, dan dikelompokkan berdasarkan tema-tema utama penelitian, yaitu konsep cloud computing, keamanan aplikasi berbasis cloud, tahapan Secure Development Lifecycle, serta kerentanan dan ancaman keamanan aplikasi. Selanjutnya, penulis melakukan analisis terhadap hubungan antara tahapan-tahapan Secure Development Lifecycle dengan kebutuhan pengamanan aplikasi berbasis cloud, serta mengkaji kontribusi setiap tahapan dalam mencegah atau memitigasi kerentanan keamanan yang umum ditemukan pada aplikasi modern. Hasil analisis kemudian disusun dalam bentuk uraian konseptual yang menekankan keterkaitan antara teori, praktik keamanan, dan kebutuhan pengembangan aplikasi cloud yang aman.

Untuk menjaga ketelitian penelitian, penulis menggunakan sumber yang berasal dari dokumen resmi, standar keamanan, dan publikasi ilmiah yang relevan dengan bidang keamanan perangkat lunak dan cloud computing. Langkah ini dilakukan agar hasil kajian memiliki dasar teoritis yang jelas dan dapat mendukung penyusunan analisis secara lebih objektif. Dengan metode tersebut, penelitian ini diharapkan mampu memberikan gambaran konseptual yang sistematis mengenai penerapan Secure Development Lifecycle dalam meningkatkan keamanan aplikasi berbasis cloud.

C. HASIL DAN PEMBAHASAN

1. Penerapan Secure Development Lifecycle pada Aplikasi Berbasis Cloud

Penerapan Secure Development Lifecycle pada aplikasi berbasis cloud dilakukan dengan mengintegrasikan praktik keamanan ke seluruh tahapan pengembangan, mulai dari perencanaan kebutuhan keamanan hingga monitoring setelah aplikasi dirilis. Microsoft menempatkan praktik seperti penetapan standar keamanan, penggunaan komponen yang aman, security design review, threat modeling, pengujian keamanan, pengamanan platform operasional, serta monitoring and response sebagai inti dari SDL. Dalam konteks aplikasi cloud, pendekatan ini menjadi penting karena keamanan tidak hanya bergantung pada source code, tetapi juga pada konfigurasi layanan, komponen pihak ketiga, pipeline deployment, dan mekanisme operasional yang berjalan terus-menerus.

Pada tahap awal, penerapan SDL dimulai dari identifikasi kebutuhan keamanan, penetapan standar, dan tata kelola keamanan yang sesuai dengan karakteristik aplikasi cloud. Tahap ini menentukan bagaimana aplikasi akan mengelola autentikasi, otorisasi, enkripsi data, logging, serta pengendalian risiko sejak fase perencanaan. Jika kebutuhan keamanan tidak didefinisikan sejak awal, maka desain dan implementasi aplikasi berpotensi menghasilkan celah yang sulit diperbaiki pada tahap akhir.

Pada tahap desain, threat modeling menjadi komponen penting dalam SDL karena membantu mengidentifikasi potensi ancaman, titik serang, alur data, dan prioritas mitigasi sebelum aplikasi dibangun. Microsoft menjelaskan bahwa threat modeling dilakukan dengan mendefinisikan komponen sistem, interaksi antarkomponen, serta data flow diagram untuk memetakan ancaman berdasarkan skenario penggunaan utama, seperti autentikasi dan pertukaran data. Dalam aplikasi berbasis cloud, threat modeling sangat relevan karena arsitektur yang terdistribusi, penggunaan API, dan ketergantungan pada layanan eksternal membuat permukaan serangan menjadi lebih kompleks.

Pada tahap implementasi, SDL menuntut penggunaan secure coding practices, penggunaan framework yang aman, serta pengendalian terhadap cryptography dan software supply chain. Tahap ini berfungsi untuk menekan munculnya kerentanan seperti injection, hard-coded secrets, autentikasi lemah, dan penggunaan komponen rentan yang sering ditemukan pada aplikasi modern. Dalam konteks cloud, pengamanan tidak cukup berhenti pada kode, tetapi juga harus mencakup dependensi, image, artefak build, dan integritas pipeline pengembangan.

Pada tahap verifikasi dan rilis, SDL menekankan pentingnya pengujian keamanan, validasi kode, serta pemeriksaan bahwa semua persyaratan keamanan telah terpenuhi sebelum aplikasi dipublikasikan. Pengujian seperti static code analysis, security testing, dan verifikasi terhadap threat model membantu menemukan kelemahan sebelum aplikasi masuk ke lingkungan produksi. Setelah aplikasi dirilis, monitoring and response menjadi bagian penting karena ancaman pada aplikasi cloud bersifat dinamis dan terus berkembang, sehingga pengamanan harus dilanjutkan melalui logging, deteksi insiden, dan proses respons yang terstandar.

2. Kontribusi Setiap Tahapan SDL terhadap Keamanan Aplikasi Cloud

Setiap tahapan SDL memiliki kontribusi yang berbeda tetapi saling melengkapi dalam meningkatkan keamanan aplikasi berbasis cloud. Tahap requirements dan governance berkontribusi dalam memastikan bahwa keamanan menjadi kebutuhan dasar sistem, bukan

tambahan di akhir pengembangan. Kontribusi terbesarnya adalah mencegah pengabaian kontrol penting seperti autentikasi kuat, kontrol akses, enkripsi, dan mekanisme audit yang dibutuhkan dalam aplikasi cloud.

Tahap desain dan threat modeling berkontribusi langsung terhadap pencegahan kerentanan kategori insecure design dan broken access control. OWASP menempatkan broken access control sebagai kategori risiko utama, sementara insecure design diperkenalkan untuk menegaskan bahwa kelemahan desain adalah akar dari banyak kerentanan aplikasi. Dengan threat modeling, pengembang dapat mengidentifikasi ancaman sejak sebelum implementasi sehingga keputusan desain lebih diarahkan pada pengurangan risiko, bukan sekadar memperbaiki dampak setelah kerentanan muncul.

Tahap implementasi berkontribusi pada pengurangan kerentanan teknis melalui secure coding practices, penggunaan komponen aman, dan pengendalian supply chain perangkat lunak. Pada tahap ini, risiko seperti injection, cryptographic failures, dan vulnerable components dapat ditekan melalui validasi input, manajemen kredensial yang baik, kontrol autentikasi, serta pemindaian dependensi dan artefak build. Dalam aplikasi cloud, kontribusi tahap implementasi menjadi sangat penting karena perubahan kode yang cepat tanpa pengamanan memadai dapat langsung berdampak pada sistem produksi.

Tahap pengujian dan verifikasi berkontribusi dalam memastikan bahwa kelemahan keamanan dapat ditemukan sebelum aplikasi dipublikasikan. Penggunaan static code analysis, vulnerability scanning, dan dynamic testing membantu mendeteksi kelemahan yang mungkin lolos pada tahap desain dan implementasi. Pada aplikasi berbasis cloud, pengujian keamanan yang terintegrasi ke pipeline sangat penting karena siklus deployment yang cepat dapat memperpendek waktu pemeriksaan manual.

Tahap deployment, monitoring, dan response berkontribusi dalam menjaga keamanan aplikasi setelah aplikasi digunakan secara nyata. Pada fase ini, SDL berfungsi untuk memastikan bahwa aplikasi dijalankan di platform yang aman, dipantau secara kontinu, serta memiliki mekanisme respons insiden jika ancaman baru muncul. Kontribusi tahap ini sangat penting di lingkungan cloud karena banyak ancaman muncul dari misconfiguration, perubahan deployment, penyalahgunaan akses, dan kegagalan monitoring yang tidak selalu terdeteksi pada tahap pengembangan.

Tahap SDL	**Risiko OWASP Top 10:2021 yang Utama Terkait**	**Contoh kontrol / aktivitas utama di tahap tersebut**

Security requirements & governance	A01: Broken Access Control, A07: Identification and Authentication Failures	Perumusan kebutuhan autentikasi dan otorisasi, penetapan kebijakan role-based access, persyaratan MFA, kebijakan pengelolaan sesi dan password.
Security design review & threat modeling	A04: Insecure Design, A05: Security Misconfiguration	Penyusunan arsitektur aman, pemodelan ancaman (threat modeling) untuk alur data dan API, penentuan trust boundary, desain defense-in-depth.
Secure coding & secure configuration	A02: Cryptographic Failures, A03: Injection	Penerapan secure coding, validasi input, penggunaan prepared statement, manajemen kunci kriptografi yang benar, konfigurasi aman komponen aplikasi.
Security testing & verification	A03: Injection, A09: Security Logging and Monitoring Failures	Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), review log keamanan, pengujian mekanisme autentikasi dan otorisasi.
Secure deployment & supply chain protection	A05: Security Misconfiguration, A08: Software and Data Integrity Failures	Review konfigurasi layanan cloud, hardening environment, verifikasi integritas artefak build, kontrol perubahan deployment, pengelolaan secret dan credential.
Monitoring, logging, dan incident response	A09: Security Logging and Monitoring Failures, A10: Server-Side Request Forgery (SSRF)	Implementasi logging terpusat, alerting, korelasi event, deteksi anomali, prosedur incident response, pemantauan akses dan trafik mencurigakan.

Table 1 Pemetaan Tahapan Secure Development Lifecycle terhadap Risiko OWASP Top 10:2021 pada Aplikasi Berbasis Cloud

Tabel ini menunjukkan bahwa setiap tahapan Secure Development Lifecycle berkontribusi pada mitigasi kategori risiko utama OWASP Top 10:2021, khususnya terkait kelemahan desain, kontrol akses, konfigurasi, dan pemantauan keamanan pada aplikasi berbasis cloud.

3. Relevansi SDL terhadap Kerentanan Umum pada Aplikasi Cloud

Secure Development Lifecycle memiliki relevansi kuat terhadap pengendalian kerentanan umum yang sering muncul pada aplikasi modern berbasis cloud. OWASP Top 10 menunjukkan bahwa kategori seperti broken access control, cryptographic failures, insecure design, dan security misconfiguration termasuk risiko yang paling dominan pada aplikasi web. Pola risiko ini menunjukkan bahwa masalah keamanan banyak berasal dari proses pengembangan yang kurang aman, bukan hanya dari serangan eksternal yang tak terduga.

Broken access control, misalnya, umumnya berkaitan dengan kelemahan dalam desain otorisasi, pembatasan hak akses, dan validasi identitas pengguna. SDL merespons risiko ini melalui penetapan security requirements sejak awal, threat modeling pada skenario autentikasi dan otorisasi, serta pengujian yang memverifikasi konsistensi kontrol akses sebelum aplikasi dirilis. Dengan demikian, SDL tidak hanya membantu mendeteksi masalah kontrol akses, tetapi juga mendorong pembentukannya secara lebih benar sejak fase desain.

Kategori insecure design juga sangat relevan karena aplikasi cloud sering dibangun secara cepat dan iteratif, sehingga keputusan desain yang lemah dapat langsung diwariskan ke proses implementasi dan deployment. Threat modeling sebagai inti SDL membantu memetakan ancaman, dependency, trust boundary, dan skenario penyalahgunaan, sehingga desain sistem dapat disesuaikan untuk meminimalkan risiko sebelum kode ditulis. Ini menunjukkan bahwa peran SDL bersifat preventif, bukan sekadar korektif.

Security misconfiguration menjadi risiko yang sangat khas pada aplikasi cloud karena banyak layanan bergantung pada konfigurasi storage, identity, network rule, secret management, dan pipeline deployment. SDL membantu menekan risiko ini melalui pengamanan engineering environment, pengamanan platform operasional, validasi deployment, dan monitoring pascarilis. Dalam konteks cloud, misconfiguration sering kali sama berbahayanya dengan bug aplikasi, sehingga pendekatan SDL menjadi relevan karena mencakup aspek operasional selain aspek coding.

Selain itu, software and data integrity failures serta kelemahan supply chain menjadi semakin penting di era cloud-native dan CI/CD. Ketergantungan pada library eksternal, container image, automation tooling, dan proses build yang terhubung menimbulkan risiko baru yang tidak dapat diatasi hanya dengan code review manual. Karena SDL memasukkan pengamanan software supply chain sebagai praktik utama, pendekatan ini lebih sesuai untuk menjawab kebutuhan keamanan aplikasi modern dibanding model pengembangan yang menempatkan keamanan hanya di fase pengujian akhir.

4. Tantangan Penerapan SDL pada Pengembangan Aplikasi Cloud

Meskipun SDL menawarkan pendekatan yang sistematis, penerapannya pada pengembangan aplikasi berbasis cloud tidak terlepas dari berbagai tantangan. Tantangan pertama adalah tekanan kecepatan pengembangan. Lingkungan cloud dan model CI/CD mendorong tim untuk merilis perubahan secara cepat, sehingga aktivitas keamanan sering dianggap memperlambat proses delivery. Jika organisasi tidak memiliki budaya secure-by-design, maka SDL berisiko dipandang sebagai beban tambahan, bukan bagian inti dari kualitas aplikasi.

Tantangan kedua adalah kompleksitas teknis pada arsitektur cloud modern. Penggunaan microservices, container, API, pipeline otomatis, serta dependensi pihak ketiga membuat ruang lingkup keamanan menjadi jauh lebih luas daripada aplikasi monolitik tradisional. Akibatnya, penerapan SDL tidak cukup hanya dengan secure coding, tetapi harus diperluas ke pengamanan konfigurasi, artefak build, supply chain, serta observabilitas sistem setelah aplikasi berjalan.

Tantangan ketiga adalah keterbatasan sumber daya dan kompetensi. Tidak semua tim pengembang memiliki kemampuan threat modeling, secure coding, code review keamanan, dan monitoring insiden yang memadai. Padahal SDL menuntut konsistensi pelaksanaan di banyak fase, sehingga tanpa pelatihan dan dukungan tata kelola yang jelas, implementasinya mudah menjadi formalitas dokumen tanpa dampak nyata terhadap keamanan aplikasi.

Tantangan keempat adalah kesulitan menjaga konsistensi implementasi keamanan dalam siklus yang berulang dan cepat. Aplikasi cloud sering mengalami perubahan konfigurasi, update dependensi, penambahan fitur, dan integrasi layanan baru yang dapat menggeser threat model awal. Jika threat modeling, security testing, dan monitoring tidak diperbarui secara kontinu, maka kontrol keamanan yang sebelumnya memadai dapat menjadi tidak relevan terhadap kondisi sistem yang terbaru.

5. Rekomendasi Penerapan SDL pada Aplikasi Berbasis Cloud

Berdasarkan hasil kajian, penerapan SDL pada aplikasi berbasis cloud akan lebih efektif jika dimulai dari integrasi keamanan pada fase requirements dan design. Tim pengembang perlu menetapkan security requirement yang jelas terkait autentikasi, kontrol akses, enkripsi, logging, dan pengelolaan data sejak awal pengembangan. Setelah itu, threat modeling perlu dijadikan aktivitas wajib untuk menilai alur data, trust boundary, potensi penyalahgunaan, dan prioritas mitigasi pada arsitektur cloud.

Rekomendasi berikutnya adalah mengintegrasikan secure coding dan security testing ke dalam pipeline pengembangan. Validasi input, kontrol autentikasi, pengamanan rahasia,

scanning dependensi, dan static analysis perlu dijalankan sebagai bagian dari proses build dan test, bukan hanya pemeriksaan manual menjelang rilis. Pendekatan ini sejalan dengan kebutuhan aplikasi cloud yang bergerak cepat dan memerlukan pemeriksaan berulang secara otomatis.

Selain itu, pengamanan deployment dan operasional harus dipandang sebagai bagian dari SDL, bukan wilayah yang terpisah dari pengembangan. Validasi konfigurasi, pengamanan artefak, pembatasan hak akses pipeline, logging terpusat, serta monitoring ancaman perlu diterapkan secara berkelanjutan agar kerentanan pascarilis dapat segera dideteksi dan direspons. Dalam lingkungan cloud, rekomendasi ini sangat penting karena perubahan kecil pada konfigurasi atau deployment dapat menimbulkan dampak keamanan yang besar.

Terakhir, organisasi perlu mendukung SDL melalui pelatihan keamanan, tata kelola yang jelas, serta evaluasi berkala terhadap implementasinya. Microsoft menempatkan training dan response sebagai aktivitas pendukung penting dalam SDL, yang menunjukkan bahwa keamanan perangkat lunak tidak hanya bergantung pada tools, tetapi juga pada kesiapan manusia dan proses organisasi. Dengan kombinasi tata kelola, otomasi, dan peningkatan kapasitas tim, SDL dapat diterapkan secara lebih realistis dan berkelanjutan pada pengembangan aplikasi berbasis cloud.

D. KESIMPULAN

Berdasarkan hasil studi literatur dan analisis konsep yang telah dilakukan, dapat disimpulkan bahwa Secure Development Lifecycle merupakan pendekatan yang relevan dan sistematis dalam meningkatkan keamanan aplikasi berbasis cloud. Penerapan SDL memungkinkan aspek keamanan diintegrasikan ke seluruh tahapan pengembangan, mulai dari penetapan kebutuhan keamanan, perancangan, implementasi, pengujian, deployment, hingga monitoring dan response. Pendekatan ini penting karena keamanan aplikasi cloud tidak hanya ditentukan oleh kualitas kode program, tetapi juga oleh keamanan desain, konfigurasi, pipeline pengembangan, komponen pihak ketiga, dan lingkungan operasional tempat aplikasi dijalankan.

Hasil pembahasan menunjukkan bahwa setiap tahapan SDL memiliki kontribusi yang saling melengkapi dalam mengurangi risiko keamanan aplikasi berbasis cloud. Tahap requirements dan governance berperan dalam memastikan keamanan menjadi kebutuhan dasar sistem, tahap design review dan threat modeling membantu mengidentifikasi potensi ancaman sejak awal, tahap implementasi dan pengujian mendukung pencegahan kerentanan

teknis, sedangkan tahap deployment, monitoring, dan response menjaga keamanan aplikasi secara berkelanjutan setelah sistem berjalan. Dengan pendekatan tersebut, SDL memiliki keterkaitan yang kuat dengan upaya mitigasi terhadap kerentanan umum pada aplikasi modern, seperti broken access control, insecure design, security misconfiguration, serta software and data integrity failures sebagaimana tercermin dalam OWASP Top 10:2021.

Di sisi lain, penerapan SDL pada pengembangan aplikasi berbasis cloud juga menghadapi sejumlah tantangan, antara lain tekanan kecepatan pengembangan, kompleksitas arsitektur cloud modern, keterbatasan sumber daya, serta perlunya konsistensi pengamanan pada setiap perubahan sistem. Oleh karena itu, penerapan SDL perlu didukung oleh tata kelola keamanan yang jelas, pelatihan yang memadai, integrasi keamanan ke dalam pipeline pengembangan, serta monitoring berkelanjutan agar implementasinya tidak hanya bersifat administratif, tetapi benar-benar berdampak pada peningkatan keamanan aplikasi. Dengan demikian, penelitian ini menegaskan bahwa Secure Development Lifecycle dapat dijadikan sebagai kerangka konseptual yang kuat untuk mendukung pembangunan aplikasi berbasis cloud yang lebih aman, proaktif, dan berkelanjutan.

E. DAFTAR PUSTAKA

Akmala, S., Riadi, I., & Prayudi, Y. (2021). Jurnal Sistem dan Teknologi Informasi Evaluasi Keamanan Sistem E-government menggunakan Security Development Lifecycle (SDL) Threat Modelling Tool. 6(2). http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO

Asrorwadi, I., Subyantoro, E., Program, D., Manajemen, S., Jurusan, I., Dan, E., Politeknik, B., Lampung, N., Soekarno-Hatta, J., 10, N., & Lampung, R. B. (2014). Design Infastructure Service as a Service (IaaS) based on Private Cloud Computing for Small Medium Entreprises Desain Model Layanan Infrastruktur Berbasis Private Cloud Computing untuk Usaha Kecil Menengah. Jurnal Ilmiah ESAI, 8(2). www.latoel.co.vu,

Satrinia Dwina, Y. S. N. M. I. M. M. (2022). Analisis Keamanan dan Kenyamanan pada Cloud Computing. 4. https://doi.org/10.52661

Sugiantoro, B. (2025). Analisis Risiko dan Solusi Keamanan Data pada Layanan Cloud Computing di Era Industri 4.0.

Vieri fadli, M., Rafly Sugiharto, M., & Nanda Saputra, M. (n.d.). EVOLUSI MEKANISME KEAMANAN DATA DALAM CLOUD COMPUTING: SYSTEMATIC LITERATURE REVIEW TERHADAP TEKNIK ENKRIPSI DAN ACCESS CONTROL (2020-2025). In Teknologi Informasi ESIT.