BAB II

LANDASAN TEORI

2.1 *Continuous Integration and Continuous Deployment (CI/CD*)

Continuous Integration (CI) adalah praktik penting dalam pengembangan perangkat lunak modern yang melibatkan penggabungan rutin perubahan kode ke dalam repositori bersama. Setiap integrasi ini diverifikasi melalui proses build dan pengujian otomatis, memungkinkan deteksi dini terhadap kesalahan atau konflik dalam kode. Pendekatan ini meningkatkan efisiensi dan kualitas perangkat lunak dengan mengidentifikasi dan mengatasi masalah sejak awal dalam siklus pengembangan (Donca dkk., 2022). Menurut penelitian (Jani, 2023), implementasi CI dapat mengurangi waktu integrasi dan meningkatkan kolaborasi tim pengembang.

Continuous Deployment (CD) melanjutkan konsep CI dengan mengotomatiskan proses pengiriman kode yang telah teruji ke lingkungan produksi. Setelah kode melewati serangkaian pengujian otomatis dan memenuhi kriteria kualitas tertentu, kode tersebut secara otomatis diterapkan ke produksi tanpa intervensi manual. Pendekatan ini memastikan bahwa fitur dan perbaikan terbaru tersedia bagi pengguna dengan cepat, meningkatkan responsivitas terhadap kebutuhan pasar. Penerapan CD dapat mengurangi risiko kesalahan manusia dalam proses rilis dan meningkatkan keandalan sistem secara keseluruhan (Istifarulah & Tiaharyadini, 2023).

2.2 Virtual Private Network (VPN)

Virtual Private Network (VPN) adalah teknologi yang memungkinkan terciptanya koneksi jaringan yang aman melalui jaringan publik seperti internet. Dengan menggunakan teknik enkripsi, VPN memastikan bahwa data yang ditransmisikan antara perangkat pengguna dan jaringan tujuan tetap rahasia dan terlindungi dari akses yang tidak sah. Menurut sebuah artikel dalam *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, VPN menyediakan koneksi aman antara pengirim dan penerima melalui jaringan publik yang tidak aman seperti internet (Sunwoo Adela Cho, 2024).

Selain itu, VPN sering digunakan untuk mengamankan komunikasi, terutama saat mengakses jaringan perusahaan dari lokasi yang berbeda, serta untuk menjaga privasi online dengan menyembunyikan alamat IP pengguna. Penelitian oleh (Hardikar dkk., 2023) menyatakan bahwa VPN membentuk terowongan virtual yang aman antara perangkat melalui jaringan publik untuk mengenkripsi lalu lintas dan menyembunyikan identitas pengguna.

2.3 Access Control List (ACL)

Access Control List (ACL) adalah mekanisme keamanan yang digunakan untuk mengatur dan mengontrol akses ke sumber daya dalam jaringan komputer. ACL terdiri dari serangkaian aturan yang menentukan izin atau penolakan akses berdasarkan alamat IP, protokol, atau parameter lainnya. Dalam konteks VPN, ACL digunakan untuk membatasi akses ke jaringan internal hanya untuk pengguna atau perangkat yang berwenang, memastikan bahwa hanya lalu lintas yang diizinkan

yang dapat melewati terowongan VPN. Menurut (Michael Oladipo Akinsanya dkk., 2024), implementasi ACL yang tepat dapat meningkatkan keamanan jaringan dengan mencegah akses yang tidak sah dan mengurangi risiko serangan.

Selain itu, ACL dapat digunakan untuk mengontrol jenis lalu lintas yang diizinkan atau ditolak dalam jaringan, seperti membatasi akses ke aplikasi tertentu atau memblokir protokol yang tidak aman. Sebuah penelitian (Tian & Gao, 2024) menyoroti bahwa penggunaan ACL yang efektif dapat meningkatkan kinerja jaringan dengan mengurangi beban lalu lintas yang tidak perlu dan meningkatkan efisiensi operasional.

2.4 CI/CD Platform

Jenkins adalah alat otomasi open-source yang digunakan untuk implementasi Continuous Integration dan Continuous Deployment dalam pengembangan perangkat lunak. Jenkins menyediakan platform yang kuat untuk mengotomatisasi berbagai tugas dalam siklus hidup pengembangan perangkat lunak, termasuk build, pengujian, dan deployment(Teixeira dkk., 2020). Menurut (Bass, 2018), Jenkins telah menjadi salah satu alat CI/CD yang paling populer karena fleksibilitas dan kemampuannya untuk diintegrasikan dengan berbagai alat dan layanan lainnya.

Selain itu, Jenkins mendukung berbagai plugin yang memungkinkan penyesuaian dan perluasan fungsionalitasnya sesuai dengan kebutuhan spesifik proyek. Sebuah penelitian (Abhishek dkk., 2022; Yang dkk., 2020) menyoroti bahwa penggunaan Jenkins dapat meningkatkan efisiensi dan konsistensi dalam

proses pengembangan perangkat lunak dengan mengurangi intervensi manual dan meningkatkan otomatisasi.

2.5 Router-Based VPN

Router-Based VPN adalah metode pengamanan koneksi jaringan yang dilakukan langsung dari perangkat router, yang berfungsi sebagai endpoint VPN dan menyediakan akses aman bagi seluruh jaringan lokal di belakangnya. Dalam konteks kerja jarak jauh (remote collaboration), router-based VPN memungkinkan tim yang berada di lokasi berbeda untuk terhubung ke jaringan kantor pusat seolah-olah mereka berada dalam satu jaringan fisik yang sama (Sunwoo Adela Cho, 2024). Dengan begitu, kolaborasi antar tim dapat dilakukan secara real-time tanpa harus mengorbankan keamanan jaringan atau integritas data yang ditransmisikan.

Keunggulan *Router-Based VPN* dalam mendukung remote collaboration terletak pada kemampuannya untuk menyediakan koneksi yang stabil, aman, dan terpusat (Michael Oladipo Akinsanya dkk., 2024). Ketika tim dari berbagai lokasi bekerja bersama menggunakan sistem kolaboratif seperti server git, file sharing, atau sistem berbasis intranet, router VPN memungkinkan mereka untuk mengakses sumber daya internal perusahaan tanpa konfigurasi tambahan pada tiap perangkat. Hal ini menyederhanakan pengelolaan akses dan mendukung kolaborasi lintas lokasi dengan pengalaman kerja yang mulus dan minim hambatan teknis (Islam dkk., 2021).

Selain itu, penggunaan *Router-Based VPN* juga meningkatkan efisiensi komunikasi antar tim dengan sistem kolaborasi. Dengan koneksi yang terenkripsi

dan kontrol akses yang tersentralisasi, perusahaan dapat menjaga keamanan informasi sensitif yang dibagikan selama proses pengembangan Bersama (Raymond Angelo, 2019). Hal ini sangat penting untuk menjaga produktivitas tim dalam skenario kerja hybrid atau remote-first yang kini semakin umum.

2.6 Modern Kernel-Level VPN

Modern Kernel-Level VPN adalah jenis VPN yang dijalankan langsung di *level kernel* sistem operasi, bukan sebagai proses aplikasi di *user space*. Pendekatan ini memungkinkan proses enkripsi, dekripsi, dan pemrosesan paket jaringan dilakukan lebih cepat dan efisien karena dijalankan lebih dekat dengan inti sistem operasi (Donenfeld, 2017). Salah satu contoh terkenal dari kernel-level VPN modern adalah WireGuard, yang dirancang untuk kesederhanaan, kecepatan, dan keamanan tinggi. Dengan pendekatan ini, latensi koneksi dapat ditekan seminimal mungkin, serta penggunaan CPU dapat lebih efisien dibandingkan VPN tradisional yang bekerja di level aplikasi (Ostroukh dkk., 2024).

Keunggulan utama dari modern kernel-level VPN adalah performanya yang sangat baik untuk skenario koneksi jangka panjang dan lalu lintas tinggi . Karena dijalankan langsung di *kernel*, VPN ini mampu memproses ribuan koneksi simultan dengan overhead minimal, menjadikannya sangat ideal untuk tim yang bekerja dari berbagai lokasi dan perlu akses cepat serta aman ke sumber daya internal. Hal ini juga mendukung integrasi yang lebih baik dan implementasi protokolnya yang lebih ringkas serta mudah diaudit (Shehab & Alzabin, 2024). Protokol seperti WireGuard, misalnya, hanya terdiri dari beberapa ribu baris kode, jauh lebih sedikit dibandingkan IPsec atau *OpenVPN*, sehingga lebih mudah dipertahankan dan

diperiksa dari potensi celah keamanan. Dengan fitur-fitur ini, modern kernel-level VPN semakin banyak digunakan oleh organisasi yang memprioritaskan efisiensi, keamanan, dan skalabilitas dalam mendukung kerja jarak jauh maupun pengembangan sistem terdistribusi (Novianto dkk., 2022).

2.7 Zero-Configuration VPN Service

Zero-Configuration VPN Service adalah jenis layanan VPN yang dirancang untuk menghilangkan kebutuhan konfigurasi manual, baik di sisi server maupun klien. Layanan ini memungkinkan perangkat untuk langsung terhubung ke jaringan privat secara otomatis tanpa perlu menyusun pengaturan IP statis, *NAT traversal*, routing, maupun pengelolaan sertifikat yang rumit. Contoh terkenal dari pendekatan ini adalah Tailscale , yang dibangun di atas protokol WireGuard, tetapi dilengkapi dengan layanan tambahan untuk manajemen identitas, otentikasi, dan konektivitas otomatis berbasis *cloud* (Tom Fenton, 2024).

Dalam konteks *remote collaboration* dan *DevOps* modern, *zero-configuration VPN* sangat bermanfaat karena mempermudah tim yang tersebar di berbagai lokasi untuk terhubung secara aman ke jaringan yang sama, tanpa intervensi tim IT untuk setiap perangkat atau lokasi baru. Anggota tim hanya perlu masuk dengan akun yang telah diberi izin, dan mereka akan langsung mendapatkan akses ke jaringan privat organisasi, lengkap dengan kontrol akses berbasis identitas. Hal ini mempercepat onboarding anggota tim baru, memperlancar kolaborasi lintas lokasi, serta mengurangi kesalahan konfigurasi yang berpotensi menyebabkan kerentanan keamanan(Chris Voisey, 2025).

Keunggulan lain dari *zero-configuration VPN* adalah kemampuannya dalam integrasi dengan ekosistem modern seperti CI/CD *pipelines*, *cloud environment*, dan infrastruktur *hybrid*. Karena layanan ini umumnya menawarkan antarmuka manajemen terpusat dan dukungan autentikasi berbasis *OAuth* atau *SSO* (*Single Sign-On*), organisasi dapat mengontrol siapa yang bisa mengakses sumber daya tertentu dengan mudah dan cepat. Model ini sangat sesuai untuk perusahaan yang menerapkan pola kerja fleksibel atau *remote-first*, di mana kecepatan, keamanan, dan kemudahan administrasi menjadi kebutuhan utama dalam membangun jaringan kolaboratif yang efisien (Chris Voisey, 2025).

2.8 Penelitian Terkait

Tabel 2.1 merupakan hasil-hasil riset mendalam yang dilakukan untuk mengetahui penelitian terdahulu, yang menjadi acuan dalam penelitian yang dilakukan untuk mengisi kesenjangan studi yang masih ada terkait kurangnya kajian komprehensif mengenai implementasi dan evaluasi teknologi *Virtual Private Network (VPN)* dalam membangun infrastruktur jaringan tertutup pada lingkungan *Continuous Integration and Continuous Deployment (CI/CD)*, khususnya dalam menilai aspek performa, kompleksitas konfigurasi, serta skalabilitas kolaboratif dalam proses pengembangan perangkat lunak.

Tabel 2. 1Penelitian Terkait

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
1	(Kjorveziroski dkk., 2024)	*Full-mesh VPN performance evaluation for a secure edge-cloud continuum*	Bagaimana memastikan keamanan dan efisiensi komunikasi antar node di lingkungan edge-cloud yang terdistribusi.	Menggunakan berbagai vendor z*ero-configuration virtual private network* untuk menunjang efisiensi integrasi antar orkestrasi Kubernetes	Penelitian ini menekankan pentingnya penggunaan solusi VPN full-mesh yang mendukung otomasi konfigurasi dan skalabilitas tinggi sebagai fondasi untuk

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
					membangun jaringan yang aman, andal, dan fleksibel dalam lingkungan edge-cloud yang terdistribusi.
2	(Jani, 2023)	*Implementing Continuous Integration And Continuous Deployment (CI/CD) In Modern Software Development*	Bagaimana meningkatkan efisiensi dan kualitas pengembangan perangkat lunak modern.	Melakukan penerapan teknologi layanan CI/CD pada lingkungan *Software Development*	Menganalisis penerapan CI/CD dalam pengembangan perangkat lunak modern untuk meningkatkan efisiensi dan kualitas perangkat lunak.
3	(Mysari & Bejgam, 2020)	*Continuous Integration And Continuous Deployment Pipeline Automation Using Jenkins Ansible*	Bagaimana mengoptimalkan komunikasi antar klaster Kubernetes secara otomatis dan aman tanpa konfigurasi manual.	Meningkatkan efisiensi integrasi dan komunikasi antar klaster Kubernetes secara otomatis dan aman tanpa konfigurasi manual	Mendorong otomasi dan skalabilitas tinggi sebagai fondasi keamanan edge-cloud terdistribusi

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
4	(Istifarulah & Tiaharyadini, 2023)	*Devops, Continuous Integration And Continuous Deployment Methods For Software Deployment Automation*	Bagaimana mempercepat dan mengotomasi proses deployment perangkat lunak.	Integrasi dan otomasi build-test-deploy untuk mempercepat siklus rilis	Meningkatkan efisiensi dan kualitas perangkat lunak
5	(Arachchi & Perera, 2018)	*Continuous Integration And Continuous Delivery Pipeline Automation For Agile Software Project Management*	Bagaimana memastikan pipeline CI/CD berjalan konsisten di berbagai environment.	Pipeline otomatis untuk deployment yang konsisten dan cepat di berbagai environment	Efisiensi deployment meningkat secara signifikan
6	(Donca dkk., 2022)	*Method For Continuous Integration And Deployment Using A Pipeline Generator For Agile Software Projects*	Bagaimana menggabungkan CI/CD dan DevOps agar deployment lebih terstandar	Menggabungkan CI/CD dan DevOps dalam satu alur otomatis untuk deployment perangkat lunak	Proses deployment lebih terstandarisasi dan cepat
7	(Bagai dkk., 2024)	*Implementing Continuous Integration And Deployment*	Bagaimana mempercepat pengiriman model ML	Meningkatkan pengiriman perangkat lunak dalam metode Agile	Pipeline mempercepat dan menstabilkan pengiriman

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
		(CI/CD) For Machine Learning Models On AWS	secara otomatis dalam pipeline Agile.		
8	(Sunwoo Adela Cho, 2024)	*The Power Of Virtual Private Networks (VPN) In Privacy Protection*	Bagaimana meningkatkan fleksibilitas dan kecepatan setup CI/CD pipeline di environment Agile.	Menyederhanakan proses setup CI/CD dengan generator template pipeline	Otomasi CI/CD meningkat dengan pengaturan lebih fleksibel
9	(Michael Oladipo Akinsanya dkk., 2024)	*Virtual Private Networks (VPN): A Conceptual Review Of Security Protocols And Their Application In Modern Networks*	Bagaimana mengotomatiskan pelatihan dan deployment model ML dalam skala besar.	Otomatisasi training dan deployment model ML secara berkala	Efisiensi pengelolaan model ML dalam skala besar meningkat
10	(Raymond Angelo, 2019)	*Secure Protocols And Virtual Private Networks: An Evaluation*	Bagaimana melindungi komunikasi internal	Menyediakan akses terenkripsi untuk	Meningkatkan keamanan dan

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
			organisasi berbasis remote work.	komunikasi internal perusahaan	anonimitas jaringan internal
11	(Putra dkk., 2022)	*Design And Implementation Of Vpn Server Using L2tp Protocol*	Bagaimana memilih protokol VPN yang tepat untuk keamanan dan performa optimal.	Memberikan gambaran teknis dan strategis penggunaan protokol VPN modern	Pemahaman terhadap kekuatan dan kelemahan protokol VPN meningkat
12	(Rahman dkk., 2020)	*Building Lptp Vpn Network And User Management Using Fuzzy Logic Based On Age And Utilization Of The Dude For Network Monitoring*	Bagaimana membandingkan performa dan keamanan protokol VPN di berbagai kondisi jaringan.	Membandingkan performa dan keamanan VPN berdasarkan protokol yang digunakan	Menunjukkan pentingnya pemilihan protokol sesuai kebutuhan keamanan
13	(Hauser dkk., 2020)	*P4-Ipsec: Site-To-Site And Host-To-Site VPN With Ipsec In P4-Based SDN*	Bagaimana mengintegrasikan VPN IPsec dalam infrastruktur SDN untuk	Menyediakan solusi VPN berbasis L2TP untuk perusahaan menengah	Solusi efektif dalam meningkatkan keamanan komunikasi jaringan

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
			meningkatkan keamanan jaringan.		
14	(Novianto dkk., 2022)	*Implementasi Keamanan Akses Terhadap Website Menggunakan Wireguard VPN Di Routerboard Mikrotik*	Bagaimana melindungi akses website internal menggunakan VPN.	Mengoptimalkan pengelolaan user VPN berbasis karakteristik pengguna	Peningkatan efisiensi manajemen pengguna VPN
15	(Shehab & Alzabin, 2024)	*Evaluating the Effectiveness of Stealth Protocols and Proxying in Hiding VPN Usage*	Bagaimana menyembunyikan lalu lintas VPN agar tidak terdeteksi oleh firewall.	Integrasi keamanan jaringan ke dalam infrastruktur SDN modern	Meningkatkan fleksibilitas dan kontrol pada VPN berbasis SDN
16	(Ostroukh dkk., 2024)	*Enhancing Corporate Network Security and Performance: A Comprehensive*	Bagaimana meningkatkan keamanan dan performa jaringan	Mengamankan akses website internal hanya untuk pengguna terverifikasi	Menjamin koneksi terenkripsi dan terbatas

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
		Evaluation of WireGuard as a Next-Generation VPN Solution	korporat dengan VPN modern.		
17	(Azmi dkk., 2022)	*Analysis And Application Of Access Control List (ACL) Methods On Computer Networks*	Bagaimana mengatur izin akses antar segmentasi jaringan secara efisien.	Mencegah deteksi penggunaan VPN oleh sistem firewall perusahaan	VPN dapat digunakan bahkan di jaringan dengan pembatasan ketat
18	(Firmansyah & Wahyudi, 2021)	*Analisis Performa Access Control List Menggunakan Metode Firewall Policy Base*	Bagaimana mengevaluasi efektivitas ACL dalam pengaturan kebijakan firewall.	Menilai efektivitas WireGuard dalam aspek kecepatan dan keamanan	WireGuard lebih unggul dalam latensi dan efisiensi enkripsi
19	(Hasan & Dewi, 2022)	*Penerapan Metode Access Control List Pada Jaringan Vlan Menggunakan Router Cisco*	Bagaimana menerapkan ACL untuk mengamankan VLAN antar jaringan.	Mengatur izin akses secara spesifik antar segmentasi jaringan	Meningkatkan kontrol keamanan terhadap akses pengguna
20	Our Research	Studi Komparatif Teknologi Virtual	Bagaimana perbedaan pendekatan arsitektur	Melakukan analisis komparatif berbasis	Penelitian ini memetakan kesesuaian

NO	Penulis	Judul	Rumusan Masalah	Solusi yang Ditawarkan	Hasil
		Private Network Untuk Optimalisasi Workflow Integrasi Infrastruktur Lingkungan Pengembangan Perangkat Lunak Berbasis CI/CD	VPN memengaruhi kompleksitas konfigurasi, pengelolaan akses, dan stabilitas integrasi workflow CI/CD.	skenario terkontrol terhadap implementasi L2TP/IPSec, WireGuard, dan Zero-Configuration VPN (Tailscale) dalam lingkungan CI/CD.	pendekatan VPN terhadap karakter operasional organisasi dan menunjukkan bahwa abstraksi jaringan berbasis identitas memberikan integrasi CI/CD yang lebih adaptif dibandingkan pendekatan VPN berbasis topologi.

2.9 Keterbaruan Penelitian

Tabel 2.2 merupakan representatif posisi penelitian ini yang menghadirkan kebaruan dengan melakukan evaluasi komparatif terhadap beberapa teknologi *Virtual Private Network (VPN)* dalam konteks integrasi infrastruktur jaringan tertutup dengan lingkungan *Continuous Integration and Continuous Deployment (CI/CD)*. Tidak seperti studi sebelumnya yang cenderung berfokus pada aspek fungsional *Virtual Private Network* atau implementasi *Continuous Integration and Continuous Deployment* secara terpisah, penelitian ini menggabungkan kedua komponen tersebut secara sistematis untuk mengkaji performa, kompleksitas konfigurasi, dan skalabilitas

kolaboratif. Dengan pendekatan ini, penelitian memberikan kontribusi terhadap pengembangan model infrastruktur yang lebih efisien dan aman bagi proses otomasi pengembangan perangkat lunak modern.

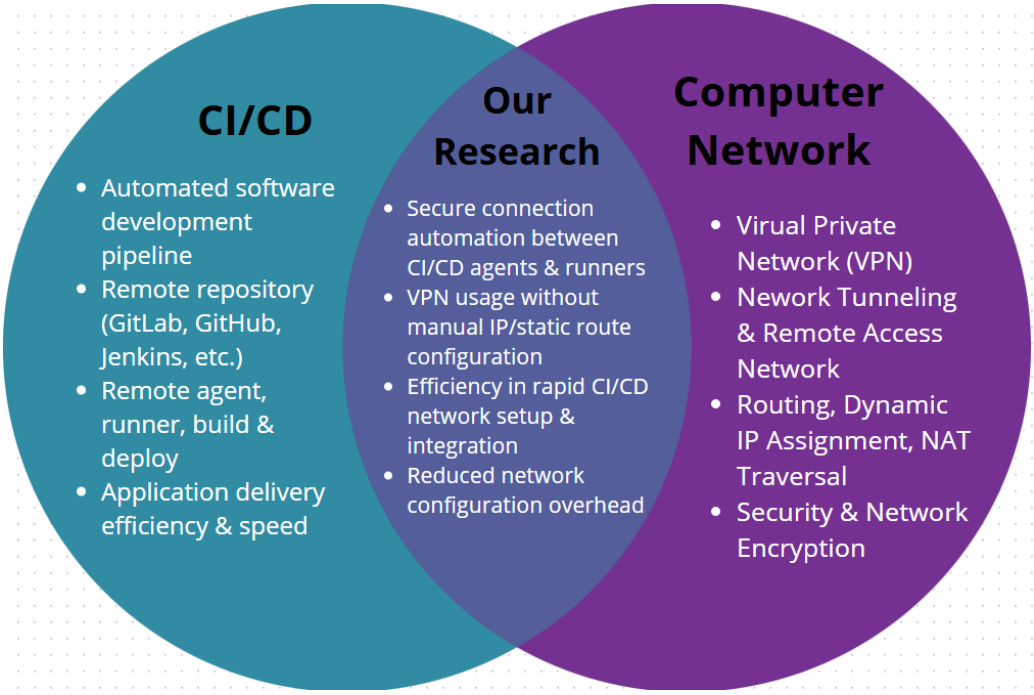
Tabel 2. 2 Matriks Penelitian

No	Penulis	Judul	CI/CD	Jenkins	Ansible	VPN	Protokol Keamanan	ACL
1	(Kjorveziroski dkk., 2024)	*Full-mesh VPN performance evaluation for a secure edge-cloud continuum*	✓	✗	✗	✓	✓	✗
2	Jani (2023)	*Implementing Continuous Integration and Continuous Deployment (CI/CD) in Modern Software Development*	✓	✗	✗	✗	✗	✗
3	Mysari & Bejgam (2020)	*Continuous Integration and Continuous Deployment Pipeline Automation Using Jenkins Ansible*	✓	✓	✓	✗	✗	✗
4	Istifarulah & Tiaharyadini (2023)	*DevOps, Continuous Integration and Continuous Deployment Methods for Software Deployment Automation*	✓	✗	✗	✗	✗	✗
5	Arachchi & Perera (2018)	*Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management*	✓	✓	✗	✗	✗	✗
6	Donca et al. (2022)	*Method for Continuous Integration and Deployment Using a Pipeline*	✓	✗	✗	✗	✗	✗

No	Penulis	Judul	CI/CD	Jenkins	Ansible	VPN	Protokol Keamanan	ACL
		Generator for Agile Software Projects						
7	Bagai et al. (2024)	*Implementing Continuous Integration and Deployment (CI/CD) for Machine Learning Models on AWS*	☑	✕	✕	✕	✕	✕
8	Sunwoo Adela Cho (2024)	*The Power of Virtual Private Networks (VPN) in Privacy Protection*	✕	✕	✕	☑	☑	✕
9	Michael Oladipo Akinsanya et al. (2024)	*Virtual Private Networks (VPN): A Conceptual Review Of Security Protocols And Their Application In Modern Networks*	✕	✕	✕	☑	☑	✕
10	(Raymond Angelo, 2019)	*Secure Protocols And Virtual Private Networks: An Evaluation*	✕	✕	✕	☑	☑	✕
11	Putra et al. (2022)	*Design And Implementation Of VPN Server USING L2TP Protocol*	✕	✕	✕	☑	☑	✕
12	Rahman et al. (2020)	*Building Lptp Vpn Network and User Management Using Fuzzy Logic Based on Age and Utilization of The Dude for Network Monitoring*	✕	✕	✕	☑	☑	✕
13	Hauser et al. (2020)	*P4-IPsec: Site-to-Site and Host-to-Site VPN With IPsec in P4-Based SDN*	✕	✕	✕	☑	☑	✕

No	Penulis	Judul	CI/CD	Jenkins	Ansible	VPN	Protokol Keamanan	ACL
14	(Novianto dkk., 2022)	*Implementasi Keamanan Akses Terhadap Website Menggunakan Wireguard VPN Di Routerboard Mikrotik*	✗	✗	✗	☑	☑	✗
15	(Shehab & Alzabin, 2024)	*Evaluating the Effectiveness of Stealth Protocols and Proxying in Hiding VPN Usage*	✗	✗	✗	☑	☑	✗
16	(Ostroukh dkk., 2024)	*Enhancing Corporate Network Security and Performance: A Comprehensive Evaluation of WireGuard as a Next-Generation VPN Solution*	✗	✗	✗	☑	☑	✗
17	Firmansyah & Wahyudi (2021)	*Analisis Performa Access Control List Menggunakan Metode Firewall Policy Base*	✗	✗	✗	✗	✗	☑
18	Hasan & Dewi (2022)	*Penerapan Metode Access Control List Pada Jaringan VLAN Menggunakan Router Cisco*	✗	✗	✗	✗	✗	☑
19	Azmi et al. (2022)	*Analysis and Application of Access Control List (ACL) Methods on Computer Networks*	✗	✗	✗	✗	✗	☑
20	Our Research	Studi Komparatif Teknologi Virtual Private Network Untuk Optimalisasi Workflow Integrasi Infrastruktur Lingkungan Pengembangan Perangkat Lunak Berbasis CI/CD	☑	☑	✗	☑	☑	☑

Gambar 2.1 merupakan diagram scope penelitian yang bertujuan untuk menggambarkan posisi penelitian yang dilakukan. Studi ini secara strategis mengisi celah antara layer aplikasi (tools CI/CD seperti GitLab/Jenkins) dan layer infrastruktur (jaringan komputer/VPN) melalui pendekatan komparatif yang mengevaluasi evolusi teknologi *Virtual Private Network* mulai dari *Virtual Private Network Router-Based* (IPSec/L2TP), *Modern Kernel-Level VPN* (WireGuard), hingga *Zero-Configuration VPN*, dengan parameter evaluasi meliputi performa, kompleksitas konfigurasi, dan skalabilitas kolaboratif. Posisi penelitian ini unik karena tidak hanya mengatasi trade-off antara kecepatan *delivery* dalam lingkungan *Continuous Integration and Continuous Deployment*, tetapi juga mengusung paradigma baru dalam manajemen jaringan dari konfigurasi manual menuju *zero-touch automation*, sehingga diharapkan dapat menjadi *enabler* bagi terwujudnya *pipeline* DevOps yang benar-benar otomatis dan aman.



Gambar 2. 1 Diagram Posisi Kontribusi Penelitian