

BAB II

TINJAUAN PUSTAKA

Bab ini mengulas perbandingan antara penelitian yang dilakukan dengan penelitian yang telah dilakukan sebelumnya. Perbandingan ini dilakukan untuk memberi acuan penulisan. Selain itu perbandingan ini memberikan penjelasan tentang keunikan penelitian yang ditulis.

Penelitian pertama dilakukan oleh Ramesh Ghimire yang berjudul *Deploying Software in the Cloud with CI/CD Pipelines* pada tahun 2020. Penelitian ini bertujuan untuk memberikan kesempatan kepada pengembang kecil untuk dapat mengimplementasikan penggunaan CI/CD dalam praktek *DevOps* yang dilakukan. Penelitian tersebut dilakukan dengan objek penelitian berupa aplikasi sederhana untuk menampilkan *hello, world!* melalui web dengan Flask. *CI/CD pipelines* yang dibangun menggunakan *repository* dari Bitbucket dan proses CI menggunakan Google Cloud Platform Cloud Build. Untuk proses *deployment* dilakukan dalam bentuk *container image* dengan menggunakan AWS CodeDeploy yang telah di *install* di AWS EC2. [4]

Penelitian kedua merupakan penelitian yang dilakukan oleh Tohirin, Sri Farida Utam, Septian Rheno Widiyanto, dan Widhy Al Mauludyansah. Penelitian ini memiliki judul Implementasi *DevOps* Pada Pengembangan Aplikasi *e-Skrining* COVID-19. Penelitian ini dilakukan dengan tujuan untuk adopsi *DevOps* pada studi kasus pengembangan perangkat lunak aplikasi *e-Skrining* COVID-19. Dalam penelitian ini digunakan beberapa teknologi pendukung seperti *code repository* oleh GitLab. Untuk proses CI dilakukan oleh Gitlab CI yang mana juga sekaligus menjalankan proses *deployment* dengan menggunakan skrip khusus yang telah dibuat menyesuaikan kebutuhan *deployment* pada server *on premise*. [5]

Penelitian ketiga dilakukan oleh Sendy Ferdian, Tjatur Kandaga, Andreas Widjaja, Hapnes Toba, Ronaldo Joshua dan Julio Narabel pada tahun 2021. Penelitian ini dilakukan dengan tujuan untuk memberikan desain *platform* efektif untuk CI/CD yang mengakomodasi kompilasi kode, eksekusi kode dan perilisan secara otomatis. Objek penelitian yang digunakan dalam pembangunan CI/CD

pipeline adalah aplikasi proyek mahasiswa dari Universitas Kristen Maranatha. Dalam penelitian ini digunakan beberapa teknologi pendukung seperti *code repository* oleh GitHub. Untuk proses CI dilakukan oleh Jenkins dan TeamCity. Tetapi dalam penelitian ini tidak di jelaskan dengan rinci bagaimana proses perilisan tersebut ke server. [6]

Penelitian keempat berjudul Implementasi CI/CD Dalam Pengembangan Aplikasi *Web* Menggunakan *Docker* dan *Jenkins* pada tahun 2021. Penelitian ini ditulis oleh Andrian Alperi dan Muhammad Arif Fadhly Ridha. Penelitian ini memiliki tujuan mengimplemntasikan dan menganalisa kualitas, waktu yang dibutuhkan dan proses otomasi pada pengembangan aplikasi *web* dengan metode CI/CD menggunakan *Docker* dan *Jenkins*. Objek penelitian yang digunakan dalam penelitian ini adalah Aplikasi bernama Elaeis. Dalam penelitian ini digunakan teknologi *code repository* GitHub. Untuk proses CI yang dilakukan dengan bantuan Jenkins. Perilisan yang dilakukan berbentuk *container image* namun tidak di jelaskan dengan rinci bagaimana proses perilisan tersebut ke server. [7]

Penelitian kelima dilakukan oleh Akbar Dhany Widiyanto, Benediktus Anindito dan Moh. Noor Al Azam. Penelitian berjudul *Implementation of Docker and Continuous Integration/Continuous Delivery for Management Information System Development*. Penelitian ini memiliki tujuan mengintegrasikan desain sistem yang menggunakan praktik CI/CD dengan implementasi menggunakan *Docker Container Platform*. Sistem tersebut akan mengintegrasikan antara pengembang dan memungkinkan perilisan *bug fix* dan fitur kedalam perangkat lunak secara otomatis tanpa menunggu lama. Dalam penelitian ini digunakan beberapa teknologi pendukung seperti *code repository* oleh Gitea. Untuk proses CI dilakukan oleh Jenkins yang mana juga sekaligus menjalankan proses *deployment* dengan melakukan perilisan dalam bentuk *container image* yang akan di *pull* setiap kali ada perilisan baru kedalam server. [8]

Berdasarkan penelitian terdahulu maka akan dibuat implementasi untuk mengubah proses integrasi, *deployment* dan perilisan perangkat lunak secara manual menjadi otomatis. Setelah itu akan dilakukan anaisis dampak yang di peroleh dari proses otomasi ditinjau dari segi waktu, biaya dan resiko. Harapan yang

timbul dari penelitian ini adalah, diharapkan dengan implementasi tersebut waktu untuk melakukan integrasi hingga perilsan dapat di hemat dan biaya serta resiko yang menyertai proses tersebut menjadi lebih efisien dan rendah.



Tabel 2.1 Tabel Perbandingan Penelitian

Pembanding	Ramesh Ghimire [4]	Tohirin, dkk [5]	Sendy Ferdian, dkk [6]	Andrian Alperi, dkk [7]	Akbar Dhany Widiyanto, dkk [8]	*) Vriyas Hartama Adesaputra
Judul Penelitian	<i>Deploying Software in the Cloud with CI/CD Pipelines</i>	Implementasi DevOps pada Pengembangan Aplikasi e-Skrining Covid-19	<i>Continuous Integration and Continuous Delivery Platform Development of Software Engineering and Software Project Management in Higher Education</i>	Implementasi CI/CD Dalam Pengembangan Aplikasi Web Menggunakan Docker dan Jenkins	<i>Implementation of Docker and Continuous Integration/Continuous Delivery for Management Information System Development.</i>	Implementasi Continuous Integration/Continuous Deployment Pada Microservice Kompres Gambar
Objek Penelitian	Aplikasi Web Hello World Menggunakan Flask	Aplikasi e-Skrining Covid-19	Aplikasi Proyek Mahasiswa Universitas Kristen Maranatha	Aplikasi Elaeis	Aplikasi Manajemen Sistem Informasi Universitas Narotama	Aplikasi Microservice Untuk Kompresi Gambar
Tujuan Perilisan Aplikasi	Server Linux AWS EC2 Instance	Server Linux <i>On Premise</i>	Tidak melakukan perilisan aplikasi	Server Linux <i>On Premise</i>	Server Linux <i>On Premise</i>	Server Linux Microsoft Azure Virtual Machine

Bentuk rilis aplikasi	<i>Container Image</i> dengan AWS CodeDeploy untuk perilsan otomatis	<i>Source Code</i> dengan skrip khusus untuk perilsan otomatis	Tidak melakukan perilsan aplikasi	<i>Container Image</i> tanpa menjelaskan bagaimana proses perilsan otomatis	<i>Container Image</i> dengan Jenkins untuk perilsan otomatis	<i>Container Image</i> dengan ArgoCD untuk perilsan otomatis
Teknologi pendukung yang digunakan	BitBucket, GCP Cloud Build, AWS CodeDeploy, AWS dan Docker	GitLab sebagai <i>code repository</i> , Gitlab CI sebagai proses CI dan CD	GitHub, Jenkins dan TeamCity	Git, Jenkins dan Docker	Gitea, Jenkins dan Docker	GitHub, Drone.io, ArgoCD, Harbor, Minio, Kubernetes
Tujuan	Memberikan kesempatan kepada pengembang software terutama yang memiliki tim kecil untuk dapat mengimplementasi penggunaan <i>CI/CD Pipelines</i> dan praktik <i>DevOps</i>	Adopsi DevOps pada pengembangan aplikasi e-Skrining Covid-19 dengan menggunakan GitLab.	Memberikan desain platform efektif untuk <i>continuous integration</i> dan <i>continuous delivery</i> untuk mengakomodasi kompilasi kode, analisis kode, eksekusi kode dan perilsan secara otomatis.	Melakukan implementasi dan menganalisa kualitas, waktu yang dibutuhkan dan proses otomasi pada pengembangan aplikasi <i>web</i> dengan metode <i>CI/CD</i> menggunakan <i>Docker</i> dan <i>Jenkins</i>	Penelitian ini memiliki tujuan mengintegrasikan desain sistem yang menggunakan praktik <i>CI/CD</i> dengan implementasi menggunakan <i>Docker Container Platform</i>	Mengembangkan implementasi <i>CI/CD Pipelines</i> terhadap <i>microservice</i> kompres berkas menggunakan Drone CI dan ArgoCD.