

BAB II

TINJAUAN PUSTAKA

2.1 Dasar Teori

Sub Bab Ini menjelaskan teori yang mendasari analisis dan definisi kebutuhan pada bab selanjutnya sehingga hasil analisis memiliki dasar yang jelas.

2.1.1 Continuous Integration

Continuous integration (CI) mengintegrasikan kode ke dalam repositori kode dan menjalankan pengujian secara otomatis, cepat, dan sering. Pengguna dapat menjalankan CI ini dengan perintah komit. (Wahyu and Guna Noviantama 2021)

2.1.2 Continuous Deployment

Continuous Deployment (CD) adalah praktek dilakukan setelah proses CI selesai dan semua kode berhasil diintegrasikan sehingga aplikasi dapat dibangun dan dipublikasikan secara otomatis. (Jaeni, S., and Laksito 2022)

2.1.3 STLC (Software Testing Life Cycle)

Pengujian perangkat lunak atau *software testing* adalah metode yang dilakukan perusahaan untuk memverifikasi apakah suatu aplikasi memenuhi persyaratan yang diharapkan, dan pengujian perangkat lunak juga digunakan untuk memastikan bahwa produk yang diproduksi bebas dari kerusakan dan cacat.

Metode ini melibatkan proses pemeriksaan komponen dan sistem perangkat lunak menggunakan alat manual dan otomatis. Pengujian perangkat lunak telah ada sejak lama, tetapi beberapa pengembang lebih suka melihatnya sebagai salah satu tugas yang perlu mereka lakukan.

Tujuan sebenarnya dari pengujian perangkat lunak atau *software testing* adalah agar perusahaan dapat mendeteksi bug dan fitur yang tidak memenuhi persyaratan sebenarnya, dan jika terdapat bug atau kesalahan pada perangkat lunak,

perusahaan harus memperbaikinya lagi dalam proses pengembangan.(Solution and Arfan n.d.)

2.1.4 Gitlab

Pengguna juga perlu memahami beberapa terminologi. *Developer* akan sering melihat istilah seperti Git, Gitlab, GitHub, Bitbucket. Mari kita lihat definisi dari semua istilah ini.

Git adalah sistem kontrol versi kode sumber yang memungkinkan Anda melacak perubahan secara lokal dan mendorong atau menarik perubahan dari sumber daya jarak jauh.

GitLab, GitHub, dan Bitbucket: Layanan yang menyediakan akses jarak jauh ke repositori Git. Layanan ini tidak hanya menghosting kode *developer*, tetapi juga menawarkan fitur tambahan yang dirancang untuk mengelola siklus pengembangan perangkat lunak Anda. Fitur tambahan ini termasuk mengelola berbagi kode antara pengguna yang berbeda, pelacakan bug, bagian wiki, dan alat pengkodean sosial lainnya.

GitLab adalah layanan mirip github yang dapat digunakan organisasi untuk menyediakan manajemen internal repositori Git mereka. Ini adalah sistem manajemen repositori Git yang dihosting sendiri yang menjaga kerahasiaan kode Anda dan memungkinkan Anda menerapkan perubahan kode dengan mudah.(Wahyu and Guna Noviantama 2021)

2.1.5 Alibaba Cloud Server

Alibaba Cloud adalah perusahaan komputasi dan layanan cloud yang dirancang untuk bisnis online dan ekosistem e-commerce. Operasi internasional Alibaba terdaftar di Singapura. Markas besar di China.

Pada tahun 2018, Alibaba membuka pusat data pertamanya di Indonesia. Peluncuran ini akan diikuti oleh yang kedua di tahun 2019 dan yang ketiga di awal

tahun 2021. Tujuan dari pusat sanitasi data adalah untuk melindungi pelanggan alibaba di Indonesia.

Pusat scrubbing data ini mendeteksi, menganalisis, dan menghapus lalu lintas berbahaya untuk melindungi sektor game dan keuangan dari serangan cyber.(Iqbal and Ashry Sallatu 2022)

2.1.6 Nginx

Nginx adalah server web dengan kinerja andal dan fitur canggih lainnya yang mudah dikonfigurasi. Hasilnya, Nginx dapat membuat situs web pengguna lebih bertenaga dan canggih.

Awalnya, Nginx hanya digunakan untuk server HTTP. Seiring perkembangan teknologi, Nginx kini juga digunakan sebagai cache HTTP, penyeimbang beban (HTTP, TCP, dan UDP), dan server proxy (IMAP, POP3, dan SMTP).

Selain fitur-fitur yang disebutkan di atas, Nginx juga dapat berjalan di berbagai sistem operasi seperti Linux, Mac OS X, HP-UX, varian BSD, dan Solaris. Nginx digunakan oleh banyak perusahaan besar termasuk Atlassian, Intuit, T-Mobile, GitLab, Microsoft, DuckDuckGo, IBM, Google, Adobe, Salesforce, Xerox, VMWare, LinkedIn, Cisco, Facebook, Twitter, Apple dan banyak lagi.(Chandra 2019)

2.1.7 Docker

Docker adalah paket perangkat lunak yang berisi semua dependensi yang diperlukan untuk menjalankan aplikasi tertentu. Semua konfigurasi dan prosedur untuk memulai atau menghentikan container diatur oleh komponen yang disebut image Docker.

Wadah atau container ini menghindari masalah kompatibilitas yang dapat terjadi saat aplikasi Anda berjalan di sistem yang berbeda. Karena dengan penampung ini perangkat lunak berjalan sama di semua lingkungan.

Container baru dibuat setiap kali pengguna menjalankan image. Mengelola wadah juga mudah dilakukan dengan Docker API atau antarmuka baris perintah (CLI). Jika diinginkan beberapa kontainer, pengguna dapat mengontrolnya menggunakan alat pembuat Docker.(Fadlulloh Romadlon Bik 2017)

2.1.8 Kubernetes

Kubernetes adalah platform open source untuk mengelola beban kerja aplikasi dalam container, menawarkan konfigurasi dan otomatisasi deklaratif. Kubernetes berada dalam ekosistem yang besar dan berkembang pesat. Layanan, dukungan, dan alat Kubernetes tersedia secara luas.

Google merilis Kubernetes sebagai proyek open source pada tahun 2014. Kubernetes dibangun berdasarkan pengalaman puluhan tahun Google dalam menjalankan beban kerja paralel, dengan ide-ide terbaik yang disumbangkan oleh komunitas.(Jahiduddin, Pramukantoro, and Bakhtiar 2020)

2.1.9 Discord

Discord adalah platform yang memungkinkan pengguna untuk mengobrol, mengirim gambar, audio, video, dan membagikan layar mereka. Discord pertama kali diterbitkan pada Mei 2015. Aplikasi ini banyak digunakan oleh pembuat konten game untuk live streaming.(Rakhmawan et al. 2020)

2.1.10 SSH

SSH adalah protokol manajemen yang memungkinkan pengguna untuk mengakses dan mengubah berbagai pengaturan dan file di server. Sudah tahu Telnet? SSH adalah evolusi Telnet dan dianggap tidak aman karena kurangnya enkripsi.

Tidak seperti Telnet, koneksi SSH dienkripsi menggunakan berbagai teknologi. B. Enkripsi Simetris, Enkripsi Asimetris, dan Hashing. Ketiganya adalah teknik kriptografi yang menjamin koneksi terenkripsi. Makanya disebut SSH, yang berasal dari kata secure shell connection atau koneksi shell yang aman. (Tohirin 2020)

2.1.11 Termius

Termius adalah klien SSH lintas platform yang berfungsi di komputer desktop dan perangkat seluler. Klien SSH ini memungkinkan Anda untuk mengatur host Anda ke dalam grup. Grup memungkinkan Anda berbagi setelan, tetapi setiap host dapat memiliki setelannya sendiri. Data ini disinkronkan dengan aman di seluruh perangkat, bersama dengan riwayat koneksi dan perintah.(Iqbal 2010)

2.1.12 Visual Studio Code

Visual Code Studio adalah editor kode gratis yang berjalan di perangkat desktop berbasis Windows, Linux, dan macOS. Editor kode ini dikembangkan oleh Microsoft, salah satu raksasa teknologi dunia. Kode Visual adalah software editor open source, namun mudah digunakan. Itu dapat digunakan untuk membuat dan mengedit kode sumber dalam berbagai bahasa pemrograman. Misalnya, JavaScript, TypeScript, Node.js, dll.(Hendri and Mochammad Arief Sutisna 2021)

2.1.13 Jira

Perangkat lunak JIRA dikembangkan oleh pengembang bernama Atlassian. Aplikasi ini awalnya dikembangkan untuk mendeteksi masalah dan kesalahan perangkat lunak. Namun lambat laun JIRA dikembangkan untuk keperluan lain dalam proses pengembangan perangkat lunak. JIRA saat ini digunakan sebagai perangkat lunak manajemen operasi.

Aplikasi ini memiliki berbagai opsi, alat, dan fitur yang dapat disesuaikan oleh pengguna untuk mengelola proyek. JIRA memungkinkan Anda untuk melakukan berbagai tugas terkait proyek, mulai dari perencanaan kerja, penjadwalan tugas, berkomunikasi dengan tim Anda, hingga melacak kendala dan kemajuan proyek.(Utomo and Prayitno 2015)

2.2 Kekurangan penerapan CI/CD Gitlab

- Strategy disiplin antara *developer* untuk identifikasi masalah harus berjalan dengan baik

- Versioning aplikasi harus diteliti, jika tidak akan membuat proses *development* jauh lebih lambat
- Tim *developer* yang cenderung tidak bisa beradaptasi pada tools-tools lain
- Table Log Erros maupun Success tidak bias di export

2.3 Kelebihan penerapan CI/CD Gitlab

- Mempercepat proses rilis produk karena penggabungan kode terjadi secara continus
- Mendeteksi bug lebih dini lewat tools CI secara otomatis
- Memperoleh feedback lebih cepat sekaligus mendeteksi eror pada proses testing
- Transparansi / visibilitas yang lebih jelas untuk mendeteksi kerusakan pada software dan mengontrol perubahan.
- SLA lebih terjaga

2.4 Metode Agile

Metode Agile adalah sebuah metodologi pengembangan perangkat lunak berdasarkan proses iteratif aturan dan solusi yang disepakati. Metode ini juga bekerja dalam sistem kolaborasi lintas tim yang terstruktur dan terorganisir..

Metode Agile sangat bagus untuk proyek jangka pendek. Karena metode ini sangat adaptif jika terjadi perubahan pada proyek. Nilai terpenting dalam agile *development* adalah tim yang dapat mengambil keputusan dengan cepat dan akurat

Keputusan tidak hanya cepat dalam waktu, tetapi juga memiliki kualitas prediktif yang baik. Memutuskan bagaimana masalah yang ada dapat diselesaikan tanpa menimbulkan masalah baru.

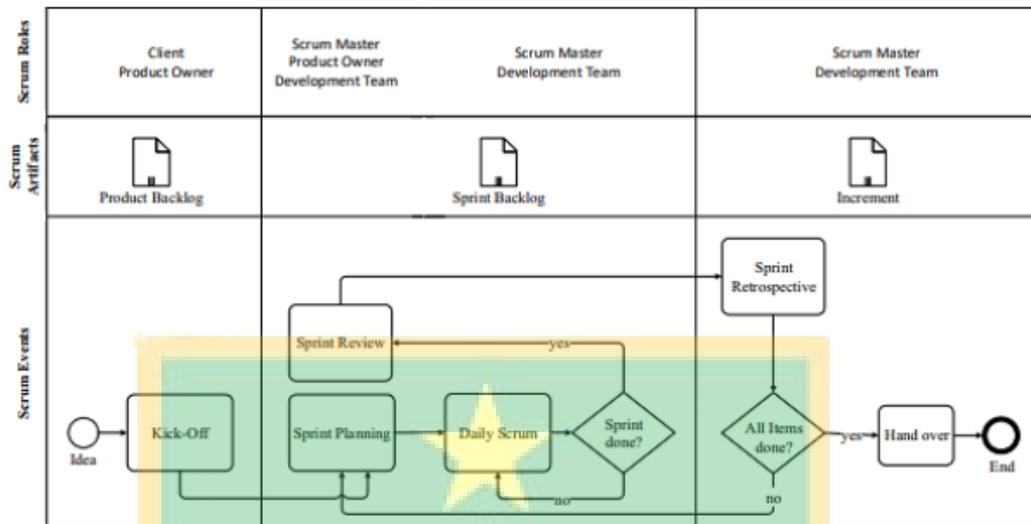
AGILE



Gambar 2. 1 Agile topology

Scrum adalah sebuah metode iteratif yang termasuk dalam metode Agile tentang cara mengelola dan menjalankan sebuah proyek. Agile adalah respons terhadap kegagalan paradigma manajemen proyek pengembangan perangkat lunak yang dominan (termasuk air terjun) dan menerapkan banyak prinsip manufaktur ramping. Pada tahun 2001, 17 pelopor metode tersebut bertemu di Resor Ski Snowbird di Utah untuk menulis Agile Manifesto. Manifesto memiliki fokus baru pada komunikasi dan kolaborasi, perangkat lunak kerja, dan organisasi tim.

Dan fleksibilitas untuk beradaptasi dengan realitas bisnis (Sachdeva, 2016). Kerangka Scrum ditunjukkan pada Gambar 2.1. Kerangka kerja Scrum terdiri dari Peran Scrum, Artefak Scrum, dan Acara Scrum (Streule, Miserini, Bartlomé, Klippel, dan De Soto, 2016), yang semuanya dijelaskan dalam beberapa bagian. Istilah scrum dipinjam dari rugby, dimana posisi dijabarkan dengan tujuan yang jelas untuk mencapai tujuan bersama (Streule et al., 2016). Selama studi ini, 10 iterasi Scrum dijalankan dengan interval dua minggu sekali.(Hadji, Taufik, and Mulyono 2019)



Gambar 2.2 Kerangka Kerja Scrum

a. Scrum Roles

Scrum Team terdiri dari Product Owner, *Development* Team, dan master scrum. Tim mengatur diri sendiri dan lintas fungsi. Pemilik Produk bertanggung jawab untuk memaksimalkan nilai proyek dan berinteraksi langsung dengan pelanggan. Individu yang melakukan pekerjaan sebenarnya adalah pengembangan. Anggota tim ini semuanya sama (tidak ada manajer proyek), mereka semua memiliki bidang keahlian masing-masing, tetapi tim memiliki tanggung jawab keseluruhan. Scrum Master memastikan bahwa setiap orang dalam tim memahami apa itu Scrum.

Salah satu tugas utama Scrum Master adalah menghilangkan hambatan yang muncul di tim Scrum. Dengan demikian pembangunan dapat fokus pada pekerjaannya dan tidak terhambat oleh hal-hal yang tidak penting. Tujuan manajemen adalah untuk memungkinkan tim Scrum menggunakan keterampilan terbaik mereka untuk mencapai tujuan mereka. (Tahyudin and Zidni Iman Sholihati 2022)

b. Scrum Artifact

Artefak Scrum dapat digambarkan sebagai elemen dengan definisi khusus di dalam kerangka kerja Scrum. Backlog produk adalah daftar prioritas produk yang berbeda (misalnya membuat denah lantai, menentukan konsep proteksi kebakaran, merancang elemen beban, dll.). Setiap item dibagi menjadi tugas yang merinci secara singkat apa yang dibutuhkan untuk pengembangan. Sprint Backlog berisi serangkaian item yang dipilih dari Product Backlog oleh Pemilik Produk dan Tim Pengembang. Kenaikan adalah jumlah dari semua produk yang diproduksi. (Tahyudin and Zidni Iman Sholihati 2022)

c. Scrum Events

Bagian ini menjelaskan berbagai aktivitas di mana Tim Scrum dapat menerapkan elemen kunci Scrum dari transparansi, inspeksi, dan kustomisasi. Diskusi awal, seperti halnya proyek apa pun diadakan berdasarkan persyaratan pelanggan dan pemilik produk membuat simpanan produk untuk memenuhi persyaratan tersebut. Ini diikuti dengan perencanaan sprint, di mana pengembangan menentukan beban kerja produk terutama dari backlog produk perencanaan poker.

Developer kemudian memilih item yang paling penting untuk digunakan dalam sprint. Selama Sprint, *Development* dan Scrum Master bertemu setiap hari untuk Daily Scrum, yang diadakan setiap hari selama Sprint pada waktu dan tempat yang sama selama 15 menit. Kemajuan dari setiap masalah yang dapat ditindaklanjuti ditampilkan di Sprint Review. Pertemuan sprint retrospective biasanya diadakan setelah sprint review. Pertemuan ini bertujuan untuk menilai secara langsung pihak-pihak yang terlibat, proses dan teknik yang digunakan, serta hubungan dan interaksinya. (Tahyudin and Zidni Iman Sholihati 2022)

2.4 Tinjauan Studi

Tinjauan pustaka adalah uraian hasil kajian pustaka yang dilakukan oleh seorang peneliti terhadap hasil penelitian yang sudah ada dan penelitian lain tentang subjek atau subjek yang serupa dengan yang diteliti atau dibahas dalam penelitian.

Di bawah ini adalah penelitian yang saya lakukan yang berkorelasi dengan penelitian yang dijelaskan dalam laporan makalah ini.:

Jurnal 1 Judul : IMPLEMENTASI CI/CD MENGGUNAKAN JENKINS PADA PEMBANGUNAN SISTEM E-POK PT PUPUK SRIWIDJAJA Penulis : Ichvandi Octa Maulana ISSN : 09021381722105 Kesimpulan Metode *microservices* menggunakan *culture* DevOps lebih mudah dan efisien dalam *menDeployment* suatu aplikasi pada saat *update* atau penambahan fitur.(Maulana 2021)

Jurnal 2 Judul: PENERAPAN CONTINUOUS INTEGRATION/CONTINUOUS DEPLOYMENT UNTUK AUTOMASI PENGUJIAN PRODUK TRAVEL STUDI KASUS: WEBSITE PEMESANAN TIKET PESAWAT PT.XYZ Penulis Nokolaus Aryawan Ravato Wijaya ISSN : 72170115 Kesimpulan : Setelah dilakukan penerapan *cutlure* atau budaya DevOps kedalam aplikasi untuk pemesanan tiket pesawat ini memudahkan para *developer* untuk melakukan *udpateing* fitur dengan waktu yang cepat, tepat dan konsisten.(Wijaya 2021)

Jurnal 3 judul :IMPLEMENTASI CONTINUOUS INTEGRATION/CONTINUOUS DELIVERY (CI/CD) PADA PERFORMANCE TESTING DEVOPS Penulis Jaeni, Nicko Aji S, Arif Dwi ISSN : 2715-3088 Kesimpulan : Dari hasil peneltian didapatkan kesimpulan bahwa proses *Deployment* menggunakan CI/CD mampu mempersingkat proses dan meningkatkan kinerja. Selain itu CI/CD memberikan hasil yang lebih teliti dengan penemuan bug pada pengujian tersebut.(Jaeni et al. 2022)

Jurnal 4 Judul: IMPLEMENTASI CONTIONOUS INTEGRATION DAN CONTINUOUS DEPLOYMENT PADA APLIKASI LEARNING MANAGEMENT

SYSTEM DI PT. MILLENNIA SOLUSI INFORMATIKA Penulis Indra Guna Noviantama, Ari Purno Wahyu W ISSN : 2407 – 3911 Kesimpulan : Dari hasil penelitian dapat disimpulkan dengan penerapan konsep CI/CD memudahkan tim pengembang dan operasional bekerja secara praktis, Dengan otomatisasi pada tahapan CI/CD ini memungkinkan kesalahan yang terjadi oleh manusia dapat dihindarkan dikarenakan seluruhnya telah otomatis dilakukan oleh mesin.(Wahyu and Guna Noviantama 2021)

Jurnal 5 Judul: IMPLEMENTASI CI/CD PIPELINE PADA FRAMEWORK ANDROBASE MENGGUNAKAN JENKINS (STUDI KASUS: PT. ANDROMEDIA) Ahmad Farid, Indra Gita Anugrah ISSN : 2621-3052 Kesimpulan : Dari hasil penelitian dapat disimpulkan bahwa dengan menerapkan sistem CI/CD pipeline dengan menggunakan aplikasi jenkins dapat mempermudah proses *Deploymen* aplikasi dimana proses *Deployment* yang awalnya menggunakan cara manual dengan CI/CD dapat dilakukan secara otomatis cara ini juga dapat menghemat waktu dan juga dapat meminimalisir kesalahan. Dengan menggunakan sistem ini semoga dapat membantu *developer* PT.Andromedia dalam mendvelop aplikasi.(Farid and Anugrah 2021)



Tabel 2.1 Tabel Perbandingan Penelitian Part 1

Pembanding	Ichvandi Octa Maulana (Maulana 2021)	Nokolaus Aryawan Ravato Wijaya [16]	Jaeni, dkk [2]
Judul Penelitian	Implementasi ci/cd menggunakan jenkins pada pembangunan sistem e-pok pt pupuk sriwidjaja	Penerapan continuous integration/continuous <i>Deployment</i> untuk automasi pengujian produk travel studi kasus: website pemesanan tiket pesawat pt.xyz	Implementasi continuous integration/continuous delivery (ci/cd) pada performance testing devops
Objek Penelitian	Aplikasi web E-POK	Aplikasi web Booking Pesawat	<i>Improvement</i> kinerja pada proses <i>Deployment</i> suatu aplikasi
Tujuan Perilisan Aplikasi	Server Linux <i>On Premise</i>	server AWS Educate	Server Linux <i>On Premise</i>
Bentuk rilis aplikasi	Jenkins CI/CD <i>CodeDeploy</i> untuk perilisan otomatis	<i>CodeDeploy</i> dengan AWS Educate, perilisan otomatis	<i>CodeDeploy</i> , Server <i>On Premise</i> , perilisan otomatis
Teknologi pendukung yang digunakan	Jenkins CI/CD, dan Server Ubuntu Linux	Jenkins CI/CD dan server VPS AWS Educate	Jenkins CI/CD dan Server Ubuntu Linux
	Menggunakan <i>culture</i> DevOps	Budaya DevOps kedalam aplikasi	Menggunakan CI/CD mampu

Tujuan	lebih mudah dan efisien dalam <i>Deployment</i> suatu aplikasi pada saat <i>update</i> atau penambahan fitur.	untuk pemesanan tiket pesawat ini memudahkan para <i>developer</i> untuk melakukan <i>update</i> fitur dengan waktu yang cepat, tepat dan konsisten	mempersingkat proses dan meningkatkan kinerja. Selain itu CI/CD memberikan hasil yang lebih teliti dengan penemuan bug pada pengujian tersebut
--------	---	---	--

Tabel 2.2 Tabel Perbandingan Penelitian Part 2

Pembanding	Indra Guna Noviantama dkk [1]	Ahmad Farid dkk [17]	*) Gunawan
Judul Penelitian	Implementasi <i>continuous integration</i> dan <i>continuous Deployment</i> pada aplikasi <i>learning management system</i> di pt. <i>Millennia</i> solusi informatika	Implementasi <i>ci/cd</i> pipeline pada <i>framework</i> <i>androbace</i> menggunakan <i>jenkins</i> (studi kasus: pt. <i>Andromedia</i>)	Penerapan <i>ci/cd</i> menggunakan arsitektur <i>microservices</i> pada aplikasi sistem e-bidan delima
Objek Penelitian	<i>Learning Management System</i> PT.	Aplikasi <i>Androbace</i>	<i>Aplikasi E-Bidan Delima Indonesia</i>

	Millennia Solusi Informatik		
Tujuan Perilisan Aplikasi	Server Linux <i>On Premise</i>	Server Linux <i>On Premise</i>	<i>Cloud base, Orcheztraction Microservices</i>
Bentuk rilis aplikasi	Gitlab CI/CD , <i>Automation Deploy</i>	perilisan otomatis	Perilisan otomatis, Container base
Teknologi pendukung yang digunakan	Gitlab CI/CD, <i>Container base</i>	Github, Jenkins CI/CD, <i>Monolith base</i>	Gitlab CI/CD, Arsitektur <i>Microservices</i> , Kubernetes, Docker, Alibaba <i>cloud</i>
Tujuan	konsep CI/CD memudahkan tim pengembang dan operasional bekerja secara praktis, Dengan otomatisasi pada tahapan CI/CD ini memungkinkan kesalahan yang terjadi oleh manusia dapat terhindarkan	jenkins dapat mempermudah proses <i>Deploymen</i> aplikasi dimana proses <i>Deployment</i> yang awalnya menggunakan cara manual dengan CI/CD dapat dilakukan secara otomatis cara ini juga dapat menghemat waktu dan juga dapat	Untuk mempercepat proses pengembangan aplikasi pada sistem E- Bidan Delima, serta meringankan pekerjaan <i>developer/programer</i> dengan tingkat kegagalan yang lebih rendah

	dikarenakan seluruhnya telah otomatis dilakukan oleh mesin.	meminimalisir kesalahan	
--	---	-------------------------	--

Metode disebut masih kurang efektif karena hal berikut :

- Jenkins UI dan Setupnya sedikit susah
- Tidak adanya penerapan *Staging Environment & Production Environment*
- Harus membutuhkan resources server untuk Jenkins yang lumayan besar
- Di jurnal ini hanya digunakan di satu repository saja.

Maka karena alasan itu penulis menggunakan tools yang berbeda yaitu Gitlab CI dan dengan *studycase* yang berbeda dengan harapan penulis metode ini juga bisa di implementasikan ke program atau aplikasi lain nantinya.

