Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

PENERAPAN CI/CD PADA DevOps GUNA MENINGKATKAN KECEPATAN DAN KUALITAS RILIS APLIKASI

Muhammad Rifqi Anuar[1], Fitri Latifah[2*]

Program Studi Informatika [1], Program Studi Informatika[2]
Fakultas Teknologi Informasi[1], Fakultas Teknologi Informasi[2]
Universitas Nusa Mandiri[1], Universitas Nusa Mandiri[2]

mhmdrifqi.anuar@gmail.com, fitri.flr@nusamandiri.ac.id[2]

Received: 2025-04-09**. Revised:** 2025-04-20**. Accepted:** 2025-04-28**. Issue Period:** Vol.9 No.2 (2025), Pp. 737-747

Abstrak: Dalam era digital, PT. Pegadaian menghadapi tantangan dalam pengembangan aplikasi akibat proses manual yang lambat dan kurang efisien, menyebabkan konflik kode dan keterlambatan rilis. Penerapan DevOps melalui CI/CD diharapkan dapat meningkatkan kecepatan, kualitas, dan responsivitas layanan, serta mengurangi risiko kesalahan dalam pengujian dan deployment.Tujuan penelitian ini adalah menganalisis penerapan CI/CD pipeline di PT. Pegadaian untuk automation deployment, mengidentifikasi dan mengevaluasi CI/CD tools yang digunakan dalam pengelolaan pipeline, serta meneliti layanan cloud berbasis container yang diterapkan dalam infrastruktur CI/CD untuk meningkatkan efisiensi dan kualitas pengembangan perangkat lunak. Implementasi CI/CD melibatkan alat seperti GitLab untuk manajemen kode, Jenkins untuk otomasi build dan deployment, Docker untuk pembuatan container, dan OpenShift sebagai platform orkestrasi. Diharapkan, metode ini menghasilkan pipeline CI/CD yang efisien, otomatis, memfasilitasi kolaborasi tim developer, dan mengevaluasi keberhasilan deployment di berbagai lingkungan. Hasil penerapan CI/CD di PT. Pegadaian menunjukkan bahwa proses integrasi dan deployment dapat dilakukan otomatis dan efisien. Tim developer dapat dengan cepat mengidentifikasi dan memperbaiki kesalahan, meningkatkan kualitas perangkat lunak, serta meminimalkan risiko kesalahan manusia dalam pengujian dan deployment di berbagai lingkungan, termasuk development, staging, dan production.

Kata Kunci : Devops, CI/CD, Otomasi, Deployment, Pengembangan Aplikasi

Abstract: *In the digital era, PT. Pegadaian faces challenges in application development due to slow and inefficient manual processes, causing code conflicts and release delays. The implementation of DevOps through CI/CD is expected to improve the speed, quality, and responsiveness of services, as well as reduce the risk of errors in testing and deployment. The purpose of this research is to analyze the implementation of CI/CD pipeline at PT Pegadaian for deployment automation, identify and evaluate CI/CD tools used in pipeline management, and examine container-based cloud services applied in CI/CD infrastructure to improve the efficiency and quality of software development. CI/CD implementation involves tools such as GitLab for code management, Jenkins for build and deployment automation, Docker for containerization, and OpenShift as an orchestration platform. Hopefully, this method results in an efficient, automated CI/CD pipeline, facilitates developer team collaboration, and evaluates deployment success in various environments. The*

DOI: 10.52362/jisamar.v9i2.1851

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

results of CI/CD implementation at PT Pegadaian show that the integration and deployment process can be automated and efficient. The development team can quickly identify and fix errors, improve software quality, and minimize the risk of human error in testing and deployment in various environments, including development, staging, and production.Devops, CI/CD, Automation, Deployment, Application Development

Keywords: *component; formatting; style; styling; insert (Minimum 3 to 5 key words)*

I. PENDAHULUAN

Pada era digital yang makin berkembang perusahaan dituntut untuk dapat merespons kebutuhan pasar dengan cepat dan efisien. Persaingan pasar yang ketat mengharuskan perusahaan untuk mempertahankan keunggulan melalui waktu pengiriman yang cepat, yang sering kali bertentangan dengan kebutuhan akan pengujian dan jaminan kualitas [1]. Namun, banyak organisasi menghadapi kendala karena penolakan terhadap perubahan budaya, perbedaan tujuan tim, dan kurangnya pemahaman lintas fungsi. Hal ini merupakan masalah yang signifikan, terutama di perusahaan besar dengan struktur tim yang lebih kaku [2]. DevOps, sebagai suatu pendekatan yang mengintegrasikan pengembangan (development) dan operasional (operations), telah muncul sebagai solusi untuk tantangan ini. Penerapan Continuous Integration (CI) dan Continuous Delivery (CD) dalam konteks DevOps menjadi kunci untuk meningkatkan kecepatan dan kualitas rilis produk atau aplikasi. Penerapan CI/CD telah terbukti mempercepat rilis produk sekaligus mengurangi risiko kesalahan dan medeteksi bug dengan cepat pada perangkat lunak [3]. Pada kesempatan ini peneliti mengambil objek penelitian di PT Pegadaian Jakarta, PT. Pegadaian sebagai salah satu perusahaan terkemuka di Indonesia, yang menawarkan berbagai layanan keuangan, seperti gadai, pembiayaan, dan investasi, bergantung pada aplikasi digital untuk menyediakan layanan kepada pelanggan. Namun PT Pegadaian menghadapi kesulitan dalam mengelola proses pengembangan aplikasi yang melibatkan banyak tim dan integrasi kode yang rumit. Hal ini sering kali memicu konflik antar kode, keterlambatan, atau ketidakstabilan aplikasi. Siklus pengembangan yang lama karena pengujian manual dan koordinasi yang tidak efisien membuat rilis fitur baru membutuhkan waktu lebih lama. Kondisi ini membuat PT Pegadaian berisiko kehilangan kesempatan pasar. Pengujian manual aplikasi yang tidak efisien dapat menyebabkan bug atau masalah performa hanya terdeteksi setelah aplikasi diluncurkan serta dapat merusak pengalaman pengguna. Proses manual dalam pengujian, tinjauan kode, dan deployment cenderung memakan waktu dan rentan terhadap kesalahan tim pengembangan atau developer. Kesalahan kecil dalam proses ini bisa mengakibatkan kerugian besar. Dalam menghadapi peningkatan jumlah pengguna dan layanan, PT. Pegadaian memerlukan sistem yang fleksibel dan mudah ditingkatkan. Sistem lama yang kaku menyulitkan tim untuk merespons perubahan kebutuhan bisnis dengan cepat. Kerja sama yang terpisah antara pengembang, penguji, dan operasional sering kali menyebabkan miskomunikasi dan kurangnya koordinasi, yang menghambat proses pengembangan.

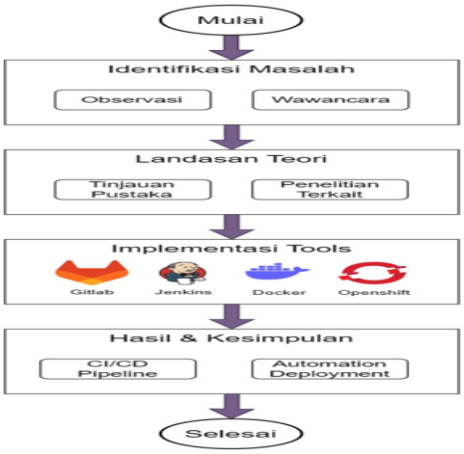
Dengan mengatasi permasalahan ini, maka diterapkanlah DevOps melalui penerapan CI/CD di PT Pegadaian, sehingga dapat mempercepat proses pengembangan, meningkatkan kualitas aplikasi, mengurangi risiko, dan memberikan layanan yang lebih responsif kepada pelanggan.

II. METODE DAN MATERI

2.1. Metode dan Tahapan Penelitian

DOI: 10.52362/jisamar.v9i2.1851

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)



Gambar 2.1. alur tahapan penelitian

Sumber : hasil penelitan tahun2024

Tahapan penelitian secara sistematis di mulai dari identifikasi masalah hingga hasil dan kesimpulan

1. Identifikasi Masalah
   a. Melakukan observasi terhadap proses penerapan CI/CD
   b. Wawancara dengan tim Release sebagai pemegang kendali serta eksekutor pada penerapan CI/CD
2. Landasan Teori
   a. Tinjaun pustaka melalui beberapa sumber seperti buku, jurnal atau artikel ilmiah yang berkaitan dengan penelitian yang berkaitan dengan penelitian.
   b. Mempelajari penelitian terkait yang serupa untuk dijadikan acuan.
3. Implementasi *CI/CD Tools*.
   Tahap ini melibatkan penerapan alat dan teknologi yang digunakan di PT. Pegadaian untuk menciptakan penerapan CI/CD dan automation deployment. Alat yang digunakan adalah sebagai berikut :
   a. *GitLab:* Sebagai platform untuk manajemen kode sumber dan integrasi dengan pipeline CI/CD.
   b. *Jenkins*: Sebagai alat otomasi untuk membangun, menguji, dan men-deploy aplikasi.
   c. *Docker:* Untuk membuat container image yang memungkinkan aplikasi dijalankan dengan konsistensi di berbagai lingkungan.
   d. *OpenShift:* Sebagai platform orkestrasi untuk deployment aplikasi ke lingkungan development, staging, ataupun production.
4. Hasil dan kesimpulan
   Hasil dari implementasi penerapan CI/CD di PT. Pegadaian adalah pipeline yang berjalan dengan indikator sebagai berikut :
   a. Jenkins dapat menjalankan pipeline secara otomatis.
   b. Menjembatani tim Developer untuk men-deploy suatu aplikasi
   c. Pipeline dapat menampilkan persetujuan yang ditujukkan kepada developer atau tech lead dengan tujuan untuk melanjutkan ataupun menghentikan pipeline.
   d. Evaluasi pipeline CI/CD yang telah dibuat.
   e. Evaluasi keberhasilan automation deployment ke berbagai lingkungan (development, staging, dan production) menggunakan openshift.

2.2. Metode CI/CD

Continuous Integration (CI) merupakan praktek yang memungkinkan pengembang untuk secara berkala melakukan penggabungan kode kedalam repository bersama, pada setiap perubahan yang aka dilakukan akan diuji secara otomatis, sehingga potensi konflik integrasi dapat di identifikasi lebih awal. Sedangkan *Continuous Delivery* (DC) merupakan proses yang memastkan setiap perubahan dalam kode siap di rilis ke produksi kapan saja, tanpa memerlukan intervensi manual. Menurut Farley dan Humble CI/CD merupakan elemen utama dalam automation proses pengembangan dan deplyment sehingga mengurangi risiko kesalahan dan mempercepat siklus rilis[2].

DOI: 10.52362/jisamar.v9i2.1851

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

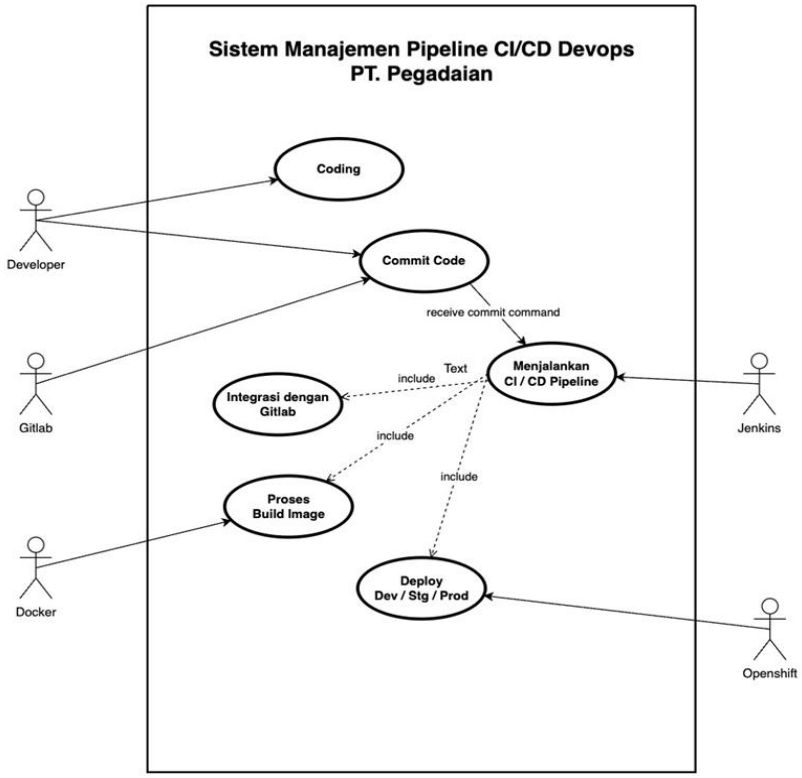
2.3. DevOps

DevOps adalah pendekatan yang menggabungkan pengembangan perangkat lunak (*development*) dan operasi (*operationt*) untuk membangun siklus pengembangan yang efisien dan responsif terhadap perubahan. Menurut Len Bass, *DevOps* dapat mengurangi hambatan komunikasi antara kedua tim ini, memungkinkan organisasi untuk beradaptasi dengan cepat terhadap kebutuhan bisnis yang dinamis serta menyediakan lingkungan pengembangan yang lebih kolaboratif dan transparan [1].

III. PEMBAHASA DAN HASIL

3.1. Use Case Penerapan CI/CD

Untuk lebih memudahkan pemahaman dalam penerapan CI/CD, peneliti membuat Use Case dari sistem penerapan CI/CD sebagai berikut



Gambar 3.1. *Use Case Diagram* Sistem CI/CD *DevOps*
Sumber Hasil Penelitian tahun 2024

Tabel 3.1. ***Deskripsi Aktor Use Case Diagram***

No	Aktor	Deskripsi
1	*Developer*	Bertugas membuat kode, melakukan *commit* dan mengelola versi dari *source code*
2	*Gitlab*	Bertugas untuk mengelola *version control*, *repository hosting*, *source code management* dan penerapan *CI/CD*

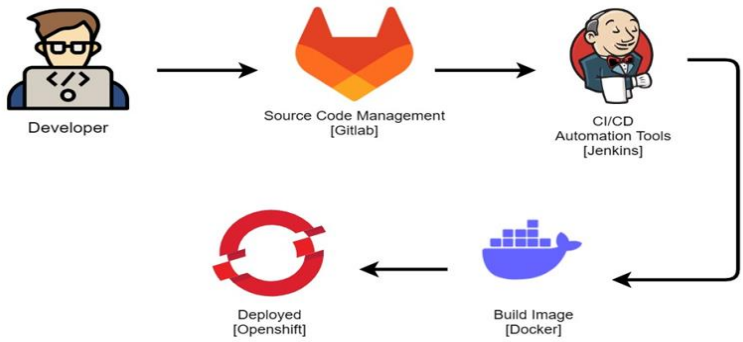
DOI: 10.52362/jisamar.v9i2.1851

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

3	*Jenkins*	Bertugas menjadi *automation tools* untuk menjalankan *CI/CD pipeline* dan pengujian otomatis
4	*Docker*	Bertugas untuk membuat, mengelola, dan menjalankan aplikasi dalam *container*
5	*Openshift*	Bertugas untuk mengelola aplikasi secara terintegrasi dengan *container* serta mengelola *deployment* aplikasi.

Sumber Hasil Penelitian tahun 2024

3.2. Perancangan Arsitektur Penerapan CI/CD DevOps



Gambar 3.2. Arsitektur Penerapan CI/CD DevOps

Penerapan arsitektur CI/CD ini melibatkan sejumlah tools yang saling berintegrasi untuk memastikan proses berjalan otomatis dan efisien. Proses diawali oleh developer yang menggunakan tools seperti Visual Studio Code untuk menulis kode, kemudian melakukan commit dan push ke repositorI GitLab. Di GitLab, kode sumber dikelola, dan pipeline otomatis dipicu melalui webhook atau konfigurasi internal, yang mengarahkan pekerjaan ke Jenkins.. Jenkins menjadi pusat dari pipeline CI/CD, mengotomatisasi tahap-tahap penting seperti membangun aplikasi dari source code, menjalankan pengujian otomatis untuk memastikan kualitas kode, dan mengemas aplikasi ke dalam bentuk image Docker. Dalam proses ini, Jenkins terintegrasi dengan Docker, yang bertugas membuat image container berdasarkan konfigurasi seperti Dockerfile, dan menyimpan image tersebut ke container registry (misalnya Docker Hub atau registry privat). Tahap akhir melibatkan Openshift, platform container berbasis Kubernetes, yang digunakan untuk menjalankan dan mengelola aplikasi. Jenkins menginstruksikan Openshift untuk menarik image Docker dari registry, kemudian menjalankan aplikasi berdasarkan konfigurasi deployment seperti DeploymentConfig atau Service.. Hubungan antar tools ini bekerja secara sinergis, dengan GitLab sebagai pemicu awal pipeline, Jenkins sebagai alat orkestrasi utama, Docker untuk pengemasan aplikasi, dan Openshift sebagai platform eksekusi yang memastikan aplikasi berjalan dan dikelola dengan baik.

3.3. Spesifikasi Perangkat Penerapan CI/CD DevOps
Spesifikasi perangkat yang digunakan untuk menerapkan pengembagan CI/CD pada penelitian ini sebagai berikut

Tabel 3.2. Spesifikasi Perangkat Penerapan CI/CD DevOps

TOOLS	JENIS HARDWARE	KAPASITAS

DOI: 10.52362/jisamar.v9i2.1851

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

TOOLS	JENIS HARDWARE	KAPASITAS
GITLAB	CPU	8vCPU
	RAM	32 GB
	Storage	500 GB SSD
	Operating System	CentOS 7
JENKINS	CPU	8vCPU
	RAM	32 GB
	Storage	500 GB SSD
	Operating System	CentOS 7
DOCKER	CPU	8vCPU
	RAM	32 GB
	Storage	200 GB SSD
OPENSHIFT	CPU	16vCPU
	RAM	32 GB
	Storage	1 TB SSD
	Operating System	RHEL 9

Sumber Hasil Penelitian tahun 2024

3.4. Pengujian Sistem

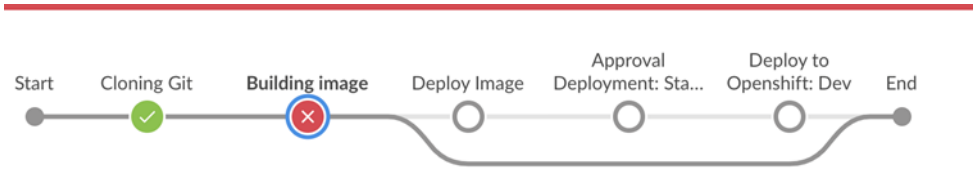
Berikut adalah pipeline yang berisi proses CI/CD pipeline pada DevOps pada penelitian yang telah dilakukan

Gambar 3.3. Proses CI/CD Pipeline berhasil di lingkungan Dev
Sumber Hasil Penelitian tahun 2024

DOI: 10.52362/jisamar.v9i2.1851

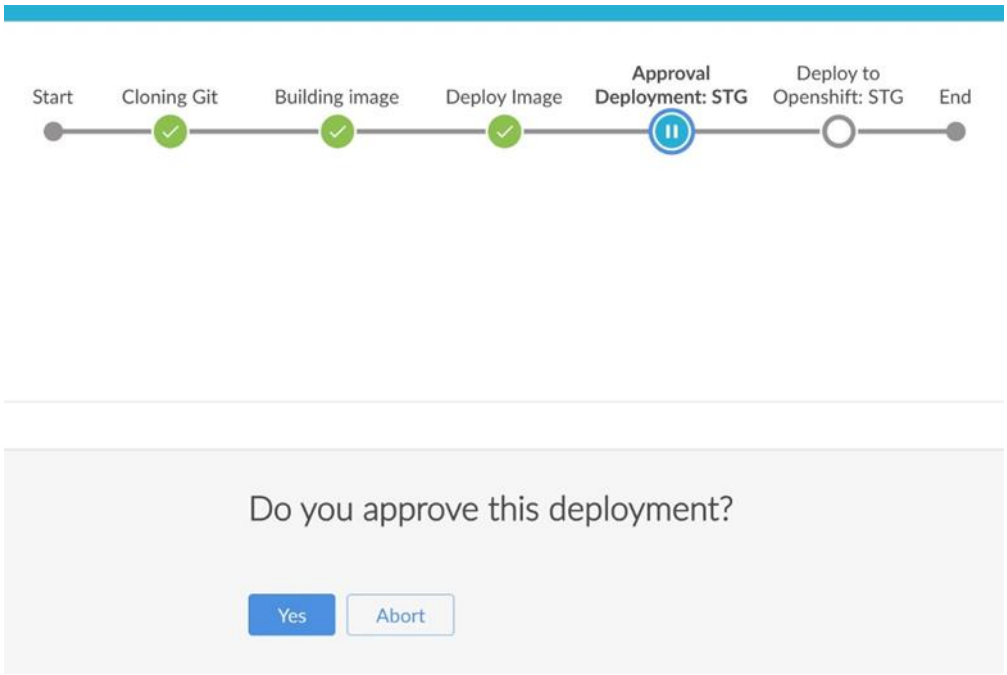
Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

, proses pipeline CI/CD berhasil melewati setiap tahapan serta proses pengujian pada pipeline hingga aplikasi ter-deploy di server Openshift Dev.



Gambar 3.4. Proses Pipeline Gagal pada stage Building Image
Sumber Hasil Penelitian tahun 2024

proses pipeline CI/CD terhenti atau error pada stage Building Image, dikarenakan ada kesalahan konfigurasi atau dependensi yang hilang pada Dockerfile dan perlu dikembalikan lagi ke Developer untuk diperbaiki.
Jika pipeline pada lingkungan Dev sudah berhasil dan disetujui oleh tech lead terkait, maka Tim Release akan melanjutkan pipeline ke lingkungan Staging seperti pada Gambar 3.5 berikut..



Gambar 3.5. proses permintaan approval pada pipeline staging

Sumber Hasil Penelitian tahun 2024

Pada pipeline di lingkungan staging, akan diminta approval maupun validasi dengan developer atau tech lead. Jika pipeline disetujui maka pipeline akan dilanjutkan hingga selesai deploy ke lingkungan staging seperti pada gambar 3.6 dan dapat berlanjut ke tahap deploy ke production.

DOI: 10.52362/jisamar.v9i2.1851

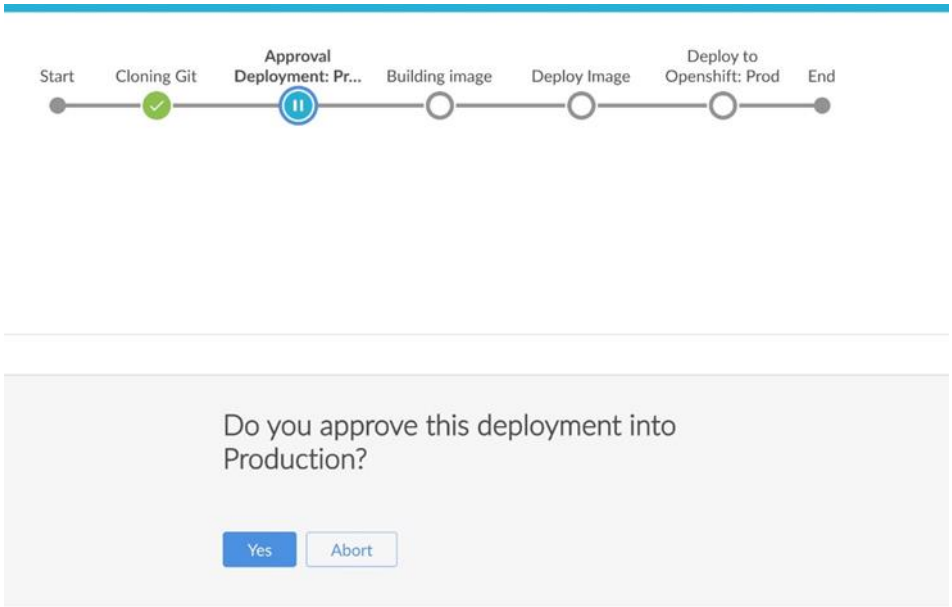
Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

Gambar 3.6 proses CI/CD Pipeline berhasil di lingkungan staging
Sumber Hasil Penelitian tahun 2024

Sedangkan jika tidak dilakukan approval maupun validasi, pipeline akan berhenti beroperasi seperti pada gambar 3.7 dan dikembalikan lagi kepada Developer agar dilakukan perbaikan pada kode terkait.

Gambar 3.7. Pipeline tidak divalidasi untuk di deploy ke lingkungan staging
Sumber Hasil Penelitian tahun 2024

Untuk pipeline production, urutan proses yang berjalan sedikit berbeda seperti pada gambar 3.8 .



Gambar 3.8 Pipeline di lingkungan production dengan approval di awal pipeline berjalan
Sumber Hasil Penelitian tahun 2024

DOI: 10.52362/jisamar.v9i2.1851

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.2 (May 2025)

Dimana permintaan approval deployment kali ini berada di awal proses pada pipeline, karena tim Release ingin memastikan bahwa kode yang sudah dikembangkan oleh Developer dan sudah melewati validasi oleh tech lead baik dari sisi lingkungan Dev maupun Staging sudah siap untuk dinaikkan ke lingkungan Production.
Pada gambar 3.9 menunujukkan bahwa proses pipeline untuk deploy ke lingkungan production sudah selesai dan aplikasi siap digunakan di lingkungan production.

Gambar3.9 Pipeline untuk deploy ke Production sudah berhasil

Sumber Hasil Penelitian tahun 2024

3.5 . Hasil Pengujian

Hasil dari penerapan CI/CD dalam DevOps pada penelitian ini menunjukkan bahwa proses pengujian dan pengintegrasian kode dapat dilakukan secara otomatis dan berkesinambungan, sehingga mempercepat siklus pengembangan perangkat lunak.

Tabel 3. 1 Pengujian Fungsional

No	Pengujian	Hasil
1	Jenkins dapat menjalankan *pipeline* secara otomatis.	Valid
2	Menjembatani tim Developer untuk men-deploy suatu aplikasi	Valid
3	Pipeline dapat menampilkan persetujuan yang ditujukkan kepada *developer* atau tech lead dengan tujuan untuk melanjutkan ataupun menghentikan *pipeline*.	Valid
4	Evaluasi *pipeline CI/CD* yang telah dibuat.	Valid
5	Evaluasi keberhasilan automation deployment ke berbagai lingkungan (development, staging, dan production) menggunakan openshift.	Valid

Sumber Hasil Penelitian tahun 2024

IV. KESIMPULAN

Berdasarkan hasil dari penelitian penerapan CI/CD dapat disimpulkan bahwa dengan penerapan CI/CD meningkatkan efisiensi proses pengebangan perangkat lunak, itegrasi otomatis dan pipeline diployment yang memungkinkan pengembang untuk medeteksi bug lebih awal sehingga waktu pengerjaan proyek akan lebih singkat. Kualitas pengujian perangkat lunak yang teritegrasi pada pipeline CI/CD yang dihasilkan memiliki kualitas yang baik dengan terlihatnya jumlah bug yang lebih kecil. Meningkat Implementasi DevOps dengan pendekatan CI/CD memfasilitasi kolaborasi yang lebih baik antara tim pengembang dan tim operasi. Proses pengembangan yang terstruktur membuat komunikasi antar tim menjadi lebih efektif.

REFERENASI

[1] L. Bass, I. Weber, and L. Zhu, *DevOps: A Software Architect's Perspective*. 2019.

DOI: 10.52362/jisamar.v9i2.1851

[2] J. Humble and D. Farley, *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2020.

[3] E. Muskardin, T. Burgstaller, M. Tappler, and B. K. Aichernig, "Active Model Learning of Git Version Control System," in *Proceedings - 2024 IEEE International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2024*, 2024, pp. 78–82. doi: 10.1109/ICSTW60967.2024.00024.

[4] P. Rai, Madhurima, S. Dhir, Madhulika, and A. Garg, "A prologue of JENKINS with comparative scrutiny of various software integration tools," in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015, pp. 201–205.

[5] C. Chen, M. Hung, K. Lai, and Y. Lin, "Docker and Kubernetes," in *Industry 4.1: Intelligent Manufacturing with Zero Defects*, IEEE, 2022, pp. 169–213. doi: 10.1002/9781119739920.ch5.

[6] G. Fontana, R. Pecora, and M. Amorim, *OpenShift Multi-Cluster Management Handbook: Go from architecture to pipelines using GitOps*. Packt Publishing, 2022. [Online]. Available: http://ieeexplore.ieee.org/document/10163101

[7] E. Wolff, *A practical guide to continuous delivery*. Addison-Wesley Professional, 2021.

[8] G. Kim, J. Humble, P. Debois, J. Willis, and N. Forsgren, *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations*. It Revolution, 2021.

[9] J. Jaeni, N. A. S., and A. D. Laksito, "IMPLEMENTASI CONTINUOUS INTEGRATION/CONTINUOUS DELIVERY (CI/CD) PADA PERFORMANCE TESTING DEVOPS," *J. Inf. Syst. Manag.*, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:252105059

[10] A. Farid and I. G. Anugrah, "Implementasi CI/CD Pipeline Pada Framework Androbase Dengan Menggunakan Jenkins (Studi Kasus: PT. Andromedia)," *J. Nas. Komputasi dan Teknol. Inf.*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:247181348

[11] A. M. Mowad, H. Fawareh, and M. Al Hassan, "Effect of Using Continuous Integration (CI) and Continuous Delivery (CD) Deployment in DevOps to reduce the Gap between Developer and Operation," *2022 Int. Arab Conf. Inf. Technol.*, pp. 1–8, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:255419041

[12] F. Zalukhu and V. Arinal, "Implementasi Sistem Persediaan Barang Berbasis Web dengan Metode DevOps pada PT. Heinz ABC Indonesia," *J. Sos. Teknol.*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:239692756

[13] M. L. Lazuardi, T. Raharjo, B. Hardian, and T. Simanungkalit, "Perceived Benefits of DevOps Implementation in Organization: A Systematic Literature Review," *Proc. 10th Int. Conf. Softw. Inf. Eng.*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:247253761

[14] P. N. Tagoug, "International Journal of Software Engineering," 2014. [Online]. Available:

DOI: 10.52362/jisamar.v9i2.1851

https://api.semanticscholar.org/CorpusID:60781506

[15] P. Lin, X. Shi, and J. Yang, "Research on the Application of DevOps in the Smart Campus of Colleges and Universities," *J. Phys. Conf. Ser.*, vol. 1883, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:235285560

DOI: 10.52362/jisamar.v9i2.1851