

Penerapan CI/CD untuk Otomasi Deployment Aplikasi Berbasis Web Menggunakan Docker dan GitHub Actions

Implementation of CI/CD for Automated Deployment of Web-Based Applications Using Docker and GitHub Actions

Asfin Rizaldy¹, Haris Yuana¹, dan Rizki Dwi Romadhona¹

¹Fakultas Teknik dan Informatika, Universitas Islam Balitar, Blitar, Indonesia

*Penulis korespondensi: rizaldyasfin@gmail.com

Abstract

Manual deployment processes in web applications often require considerable time and are prone to configuration errors. This issue is also present at SMK PGRI Wlingi, which still relies on manual deployment methods, resulting in potential delays and technical difficulties. In some cases, a single deployment may take up to two hours. This study aims to implement deployment automation by applying Continuous Integration, Continuous Delivery, and Continuous Deployment (CI/CD) using Docker and GitHub Actions to the Digiham Sarana application. The research method employed is Design Science Research (DSR), which consists of six main stages: problem identification, objective definition, design and development, demonstration, evaluation, and communication of results. After the automation system was designed and implemented, five rounds of testing were conducted to measure the average deployment duration while evaluating the speed and stability of the process. The results show that the deployment process can be executed automatically without manual intervention, achieving an average time of 4 minutes and 49 seconds. Functional testing using the blackbox method demonstrated 100% accuracy, with all three test scenarios running as expected. System evaluation was also carried out through close beta testing involving four technical users and one expert validator. The evaluation results indicate a feasibility score of 83.33% from technical users and 85.71% from the expert validator. Based on these findings, the system is considered feasible, efficient, and consistent in automating the web application deployment process within the context of SMK PGRI Wlingi.

Keywords: CI/CD, deployment automation, web application, GitHub Actions, Docker, SMK PGRI Wlingi

Abstrak

Proses deployment manual pada aplikasi web kerap memakan waktu lama dan rentan terhadap kesalahan konfigurasi. Hal ini juga terjadi di SMK PGRI Wlingi yang masih menggunakan metode manual, sehingga menimbulkan risiko keterlambatan dan kendala teknis. Dalam beberapa kasus, waktu yang dibutuhkan untuk satu kali deployment dapat mencapai dua jam. Penelitian ini bertujuan menerapkan otomatisasi deployment dengan mengimpelementasikan Continuous Integration, Continuous Delivery, dan Continuous Deployment (CI/CD) menggunakan Docker dan GitHub Actions pada aplikasi Digiham Sarana. Metode yang digunakan adalah Design Science Research (DSR), yang terdiri atas enam tahapan utama, yaitu identifikasi masalah, perumusan tujuan, desain dan pengembangan, demonstrasi, evaluasi, serta komunikasi hasil. Setelah sistem otomasi dirancang dan diterapkan, dilakukan lima kali pengujian untuk mengukur rata-rata durasi deployment, sekaligus mengevaluasi kecepatan dan kestabilan prosesnya. Hasil pengujian menunjukkan bahwa proses deployment dapat berjalan secara otomatis tanpa intervensi manual, dengan waktu rata-rata 4 menit 49 detik. Pengujian fungsional melalui metode blackbox menunjukkan tingkat akurasi sebesar 100%, dengan seluruh 3 skenario berjalan sesuai harapan. Evaluasi sistem juga dilakukan melalui tahap close beta testing yang melibatkan empat pengguna teknis serta satu validator ahli. Hasil evaluasi menunjukkan skor kelayakan sebesar 83,33% dari pengguna teknis dan 85,71% dari validator. Berdasarkan seluruh temuan tersebut, sistem dinilai layak, efisien, dan konsisten dalam mengotomasi proses deployment aplikasi web di lingkungan SMK PGRI Wlingi.

kata kunci: CI/CD, otomatisasi deployment, aplikasi web, GitHub Actions, Docker, SMK PGRI Wlingi

PENDAHULUAN

Deployment merupakan proses pengiriman aplikasi dari lingkungan pengembangan ke lingkungan produksi. Dalam praktiknya, deployment manual sering menghadapi kendala seperti waktu yang lama, risiko kesalahan konfigurasi, dan inkonsistensi lingkungan [1]. Studi sebelumnya menunjukkan bahwa pendekatan manual memiliki tingkat kegagalan lebih tinggi dibandingkan otomasi deployment [2].

Penerapan otomasi melalui pendekatan Continuous Integration, Continuous Delivery, dan Continuous Deployment (CI/CD) memungkinkan proses integrasi, pengujian, hingga deployment aplikasi dilakukan secara efisien dan minim kesalahan [3]. Salah satu penelitian menunjukkan

bahwa penerapan CI/CD menggunakan Docker dan Jenkins mampu memangkas waktu deployment dari 8 menit 48 detik menjadi 2 menit 56 detik [4].

GitHub Actions juga menjadi alternatif populer untuk CI/CD karena terintegrasi langsung dengan repositori dan dapat menurunkan waktu eksekusi hingga 33,33%, serta meningkatkan tingkat keberhasilan deployment sebesar 18,75% [5]. Di sisi lain, Docker berperan penting dalam menjaga konsistensi lingkungan aplikasi, karena aplikasi dapat dikemas dalam kontainer yang terisolasi dan identik di berbagai platform [1].

Hingga saat ini, SMK PGRI Wlingi masih menggunakan metode manual dalam proses deployment aplikasi web. Proses ini tidak hanya memakan waktu, tetapi juga

rawan kesalahan akibat konfigurasi manual. Penelitian ini melakukan simulasi penerapan CI/CD berbasis GitHub Actions dan Docker pada aplikasi Digiham Sarana, yang digunakan untuk manajemen peminjaman alat praktik di sekolah. Tujuan dari penelitian ini adalah mempercepat proses deployment dan mengurangi risiko kesalahan akibat konfigurasi manual. Hasil penelitian diharapkan dapat menjadi acuan untuk penerapan otomasi deployment di aplikasi lain dalam lingkungan pendidikan.

METODE PENELITIAN

Penelitian ini menggunakan metode Design Science Research (DSR), yaitu metode yang berfokus pada penciptaan solusi inovatif melalui desain artefak yang mampu menyelesaikan permasalahan di dunia nyata [1]. Artefak yang dikembangkan dalam DSR dapat berupa model, metode, sistem, framework, atau instrumen yang dirancang untuk menjawab masalah yang dihadapi. Tujuan utama dari pendekatan ini adalah menghasilkan design knowledge, yakni pemahaman mengenai bagaimana sebuah solusi teknologi dapat dirancang dan diterapkan secara efektif. Tahapan-tahapan DSR digambarkan pada Gambar 1.

IDENTIFIKASI MASALAH

Identifikasi masalah dilakukan melalui observasi dan wawancara terhadap proses kerja pengelola sistem aplikasi web di SMK PGRI Wlingi. Diketahui bahwa proses deployment aplikasi saat ini dilakukan secara manual, dimulai dari pull kode dari GitHub, melakukan build lokal, mentransfer file menggunakan WinSCP, hingga konfigurasi aplikasi melalui Plesk Panel. Proses ini membutuhkan waktu lama, sekitar 2 jam, dan rentan terhadap kesalahan konfigurasi akibat keterlibatan manual di setiap tahap. Beberapa kendala utama yang ditemukan meliputi:

- Tidak adanya sistem otomatisasi dalam proses deployment.
- Proses build dan pengiriman aplikasi masih dilakukan manual.
- Waktu deployment yang lama dan tidak efisien.

Dari hasil analisis, disimpulkan bahwa diperlukan sistem otomasi deployment yang mampu mengurangi ketergantungan pada proses manual, serta mempercepat waktu deployment aplikasi.

MENENTUKAN TUJUAN

Penelitian ini bertujuan untuk menerapkan sistem deployment otomatis menggunakan Continuous Integration, Continuous Delivery, dan Continuous Deployment (CI/CD) berbasis GitHub Actions dan Docker. Sistem ini dirancang untuk menggantikan proses deployment manual yang masih digunakan di SMK PGRI Wlingi, sehingga proses deployment dapat berjalan otomatis, lebih cepat, dan mengurangi potensi kesalahan.

DESAIN DAN PENGEMBANGAN

Pada tahap ini, dilakukan perancangan dan pengembangan sistem deployment otomatis berbasis CI/CD (Continuous Integration, Continuous Delivery, dan Continuous Deployment) yang diimplementasikan menggunakan GitHub Actions dan Docker. Sistem ini dikembangkan untuk menggantikan proses deployment manual yang selama ini digunakan di SMK PGRI Wlingi, sehingga proses deployment dapat berjalan lebih cepat, otomatis, dan minim kesalahan.

CI/CD merupakan pendekatan otomatisasi dalam pengembangan perangkat lunak yang mencakup integrasi kode, pengujian, pengemasan aplikasi, hingga pengiriman otomatis ke server produksi [3]. GitHub Actions digunakan sebagai platform otomasi berbasis event yang terintegrasi langsung dengan GitHub, memungkinkan penyusunan alur kerja otomatis menggunakan file YAML [3]. Sementara itu, Docker digunakan untuk mengemas aplikasi dan seluruh dependensinya ke dalam kontainer yang ringan dan konsisten, sehingga memastikan keseragaman lingkungan dari tahap pengembangan hingga produksi [4].

Alur sistem digambarkan menggunakan flowchart untuk memvisualisasikan urutan proses secara sistematis. Flowchart menyajikan langkah-langkah logis dalam bentuk simbol dan panah alur, sehingga memudahkan pemahaman proses kerja sistem [5].

Rancangan sistem mencakup tiga proses utama sebagai berikut:

Continuous Integration

Tahap ini bertugas membangun dan menguji kode aplikasi setiap kali terjadi perubahan. Proses diawali saat kode di-push ke repository GitHub, lalu GitHub Actions akan memicu workflow untuk melakukan proses build dan test. Jika proses pengujian berjalan sukses, maka sistem akan melanjutkan ke tahap berikutnya.

Continuous Delivery

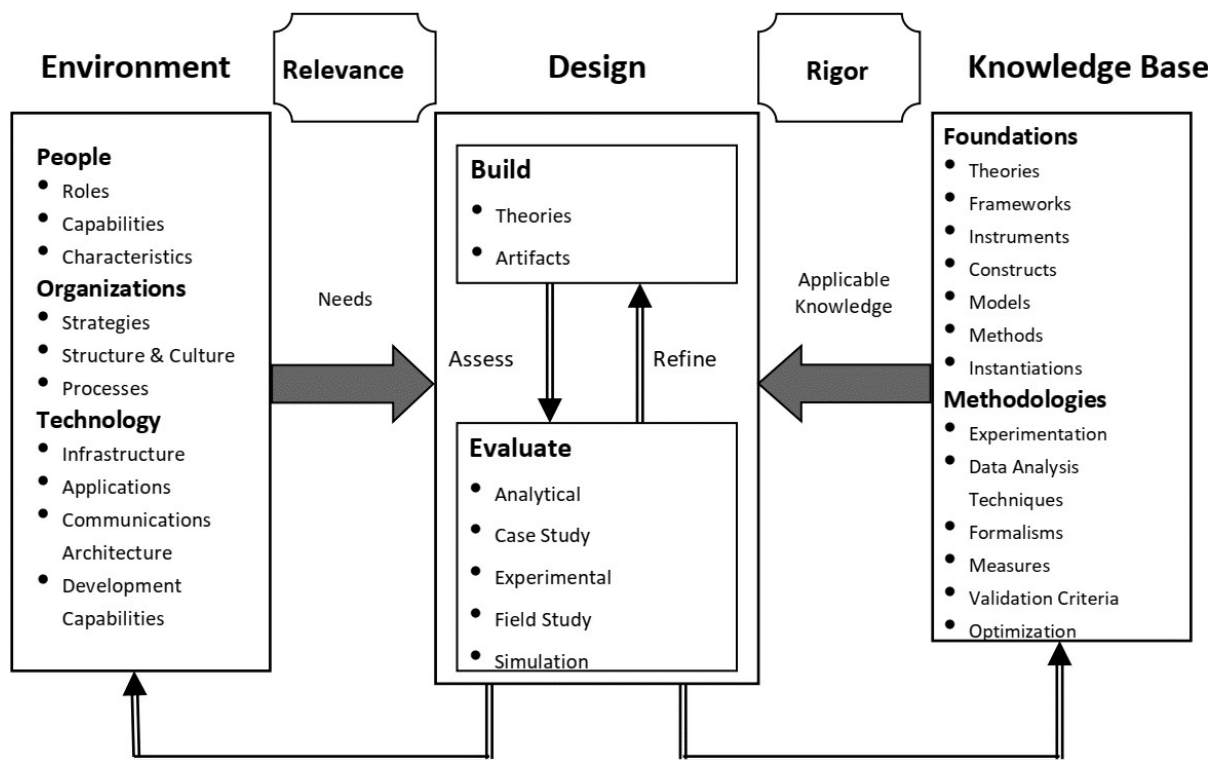
Setelah kode berhasil melalui tahap integrasi dan pengujian, artefak sistem dikemas menjadi Docker image. Image ini kemudian di-push ke GitHub Container Registry (GHCR). Proses ini juga mencakup pembuatan image untuk layanan pendukung seperti database, yang selanjutnya akan dikirim ke registri untuk siap digunakan pada tahap deployment.

Continuous Deployment

Tahapan terakhir adalah proses pengambilan dan penerapan image di server produksi. Sistem secara otomatis akan menarik image terbaru dari GHCR, lalu menjalankannya menggunakan Docker di server. Jika image berhasil dijalankan, maka aplikasi berhasil di-deploy dengan sukses.

DEMONSTRASI

Setelah proses pengembangan selesai, dilakukan tahap demonstrasi untuk menguji penerapan sistem otomasi deployment pada aplikasi Digiham Sarana. Demonstrasi dilakukan menggunakan server pribadi yang dikonfigurasi untuk mensimulasikan lingkungan operasional di SMK PGRI Wlingi.



Gambar 1. Metode Design Science Research

Tabel 1. Spesifikasi Server

No	Komponen	Spesifikasi
1.	Jenis Server	VPS (Virtual Private Server)
2.	Penyedia	DigitalOcean
3.	RAM	4 GB
4.	Penyimpanan	80 GB
5.	CPU	2 Core
6.	Sistem Operasi	Ubuntu 24.10

Pemanfaatan server pribadi dilakukan untuk keperluan simulasi sistem secara independen. Meskipun tidak menggunakan server milik sekolah secara langsung, spesifikasi yang digunakan telah disesuaikan agar merepresentasikan kondisi nyata. Adapun spesifikasi server uji dapat dilihat pada Tabel 1.

EVALUASI

Evaluasi dilakukan untuk menilai kelayakan dan keandalan sistem deployment otomatis yang telah dikembangkan. Evaluasi ini terdiri dari dua pendekatan utama, yaitu black-box testing dan close beta testing.

Blackbox testing digunakan untuk menguji apakah setiap proses dalam alur kerja CI/CD berjalan sesuai fungsi yang diharapkan tanpa melihat struktur kode internal. Pengujian dilakukan berdasarkan input dan output sistem dari sudut pandang pengguna, untuk memastikan bahwa sistem bekerja sesuai kebutuhan yang telah ditetapkan [6].

Sementara itu, close beta testing dilakukan dengan melibatkan sekelompok pengguna terbatas, terdiri dari pengguna teknis dan validator ahli. Tujuannya adalah untuk mem-

peroleh umpan balik terhadap fungsionalitas dan stabilitas sistem dalam lingkungan nyata sebelum implementasi lebih luas [8]. Penilaian dilakukan menggunakan instrumen kuesioner berbasis skala likert guna memperoleh evaluasi yang terstruktur dan obyektif.

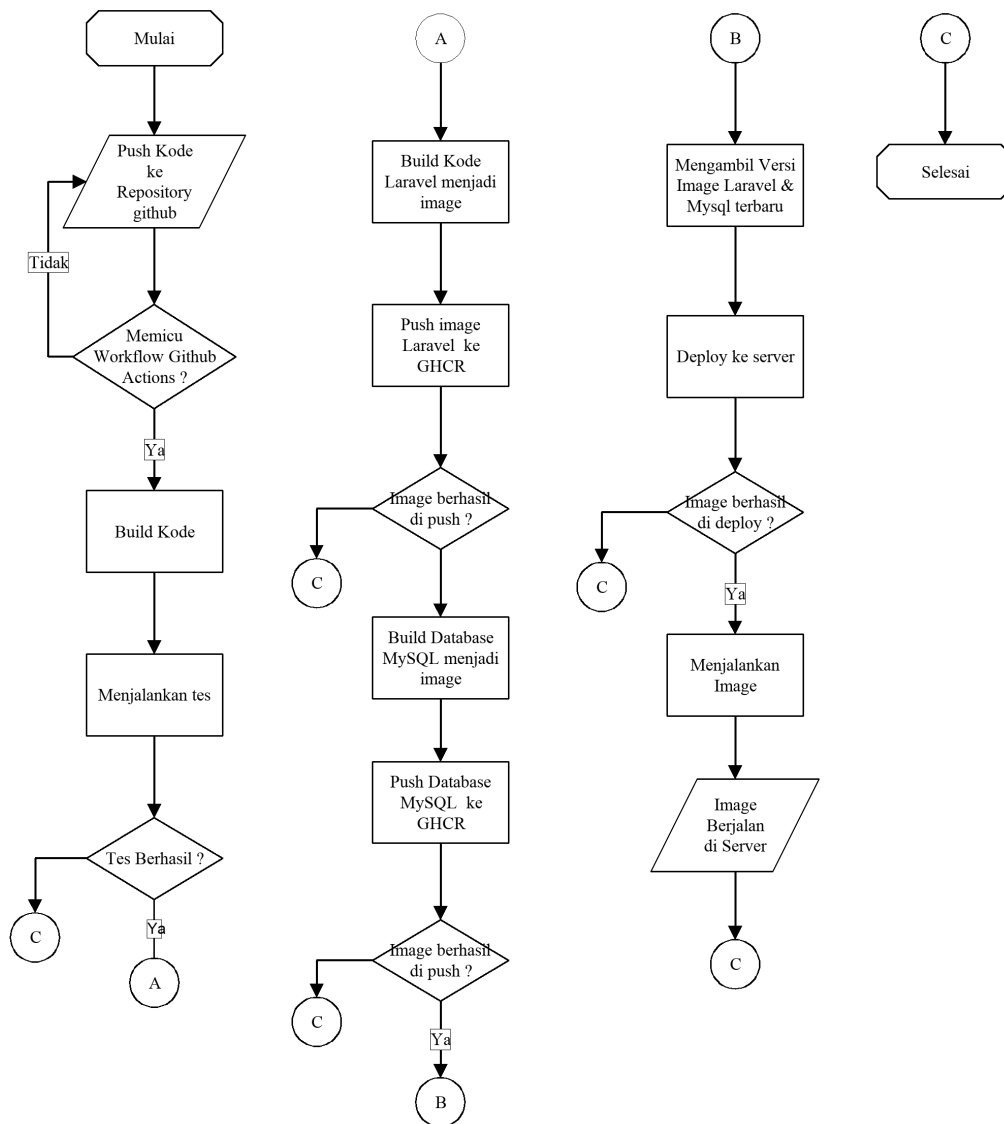
KOMUNIKASI

Tahap komunikasi dilakukan dengan menyampaikan hasil perancangan dan evaluasi sistem otomatis deployment kepada pihak terkait, khususnya pengelola sistem di SMK PGRI Wlingi. Informasi disampaikan dalam bentuk laporan dan diskusi untuk memastikan bahwa solusi yang dikembangkan dapat dipahami serta dipertimbangkan untuk diimplementasikan. Selama proses ini, pihak pengguna memberikan umpan balik terhadap sistem yang telah dibangun. Apabila ditemukan kebutuhan tambahan atau ketidaksesuaian dengan tujuan awal, maka dapat dilakukan penyesuaian kembali pada tahap sebelumnya dalam tahap Design Science Research.

HASIL DAN PEMBAHASAN

HASIL

Hasil dari penelitian ini adalah sistem otomatis deployment aplikasi web berbasis CI/CD yang dirancang untuk SMK PGRI Wlingi. Sistem ini mampu menjalankan tahapan Continuous Integration, Continuous Delivery, dan Continuous Deployment secara otomatis untuk membantu proses pengujian hingga pengiriman aplikasi ke server produksi tanpa konfigurasi manual. Ketiga tahapan ini dikonfigurasi dalam satu workflow pada GitHub Actions dan telah berhasil diterapkan untuk mendukung deployment



Gambar 2. Flowchart Sistem CI/CD

aplikasi Digiham Sarana.

PEMBAHASAN

Berikut adalah hasil yang diperoleh setelah melaksanakan tahap Desain dan Pengembangan serta Demonstrasi dalam metode Design Science Research (DSR):

Hasil Continuous Integration

Tahapan ini bertujuan untuk memastikan bahwa kode program yang dikembangkan dapat diuji secara otomatis setiap kali diperbarui. Gambar berikut menampilkan hasil proses integrasi yang meliputi pengecekan struktur kode dan pengujian otomatis terhadap fitur-fitur sistem. Semua langkah pada tahap ini berhasil dijalankan tanpa kesalahan, yang menunjukkan bahwa sistem siap untuk dilanjutkan ke tahap berikutnya (Gambar 3).

Hasil Continuous Delivery

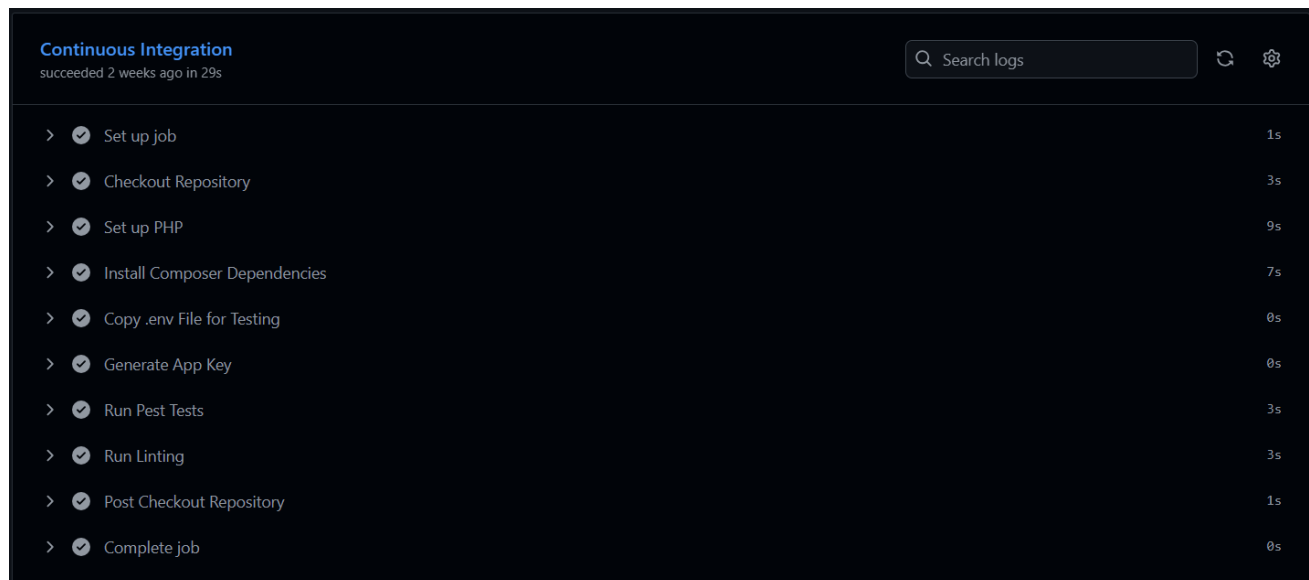
Setelah kode berhasil diverifikasi, sistem melanjutkan ke tahap pembuatan versi aplikasi yang siap dijalankan. Tahapan ini mencakup proses pembuatan paket aplikasi dalam

bentuk yang dapat digunakan di lingkungan server. Gambar berikut menunjukkan proses pembuatan dan pengunggahan paket aplikasi ke repositori, yang kemudian digunakan sebagai sumber utama dalam proses deployment (Gambar 4).

Hasil Continuous Deployment

Pada tahap ini, versi aplikasi yang telah selesai diproses secara otomatis dikirim dan dijalankan pada server tujuan. Seluruh proses dilakukan tanpa intervensi manual, dimulai dari penyiapan konfigurasi hingga pelaksanaan perintah pengiriman ke server. Tahapan ini memastikan bahwa setiap perubahan atau pembaruan aplikasi dapat langsung diterapkan ke lingkungan server dengan cepat dan konsisten. Gambar berikut memperlihatkan bahwa sistem berhasil menyelesaikan proses pengiriman dan penerapan versi terbaru aplikasi di server produksi (Gambar 5).

Untuk mengetahui waktu yang dibutuhkan dalam proses deployment, dilakukan pengujian sebanyak lima kali pada sistem otomatis deployment menggunakan CI/CD. Hasilnya disajikan pada Tabel 2.



Gambar 3. Hasil Continuous Integration

Tabel 2. Perhitungan Waktu Deployment

No	Continuous Integration	Continuous Delivery	Continuous Deployment	Total
1.	27s	3m 40s	56s	5m 3s
2.	21s	3m 19s	49s	4m 29s
3.	22s	4m 3s	42s	5m 7s
4.	23s	4m 0s	42s	5m 5s
5.	21s	3m 15s	45s	4m 21s
Rata-rata				4m 49s

Hasil menunjukkan bahwa proses deployment berlangsung cepat dan stabil, dengan rata-rata waktu 4 menit 49 detik untuk menyelesaikan seluruh tahapan CI/CD, jauh lebih singkat dibanding proses sebelumnya yang memerlukan hingga 2 jam.

Tampilan berikut menunjukkan halaman utama aplikasi Digiham Sarana yang telah berhasil dideploy dan berjalan di server (Gambar 6).

PENGUJIAN

Pengujian dilakukan pada tahap dalam metode Design Science Research (DSR). Evaluasi ini mencakup dua pendekatan, yaitu blackbox testing untuk menguji alur fungsional sistem, serta close beta testing untuk memperoleh penilaian kelayakan sistem dari perspektif pengguna dan validator ahli.

Pengujian blackbox dilakukan terhadap tiga proses utama dalam alur CI/CD, sebagaimana ditunjukkan pada Tabel 3.

Hasil pengujian menunjukkan bahwa seluruh proses berjalan sesuai harapan tanpa ditemukan kesalahan. Sistem berhasil melakukan proses CI/CD dengan tidak adanya kegagalan pada seluruh kasus uji, sistem mencapai tingkat akurasi sebesar 100%, yang mencerminkan keberhasilan penuh dalam menjalankan alur otomatisasi CI/CD.

Selanjutnya, close beta testing dilakukan terhadap lima responden, terdiri dari empat pengguna teknis dan satu validator ahli. Penilaian menggunakan kuesioner skala Likert lima poin untuk mengukur persepsi terhadap sistem. Kat-

egori kelayakan sistem berdasarkan persentase nilai dapat dilihat pada Tabel 4.

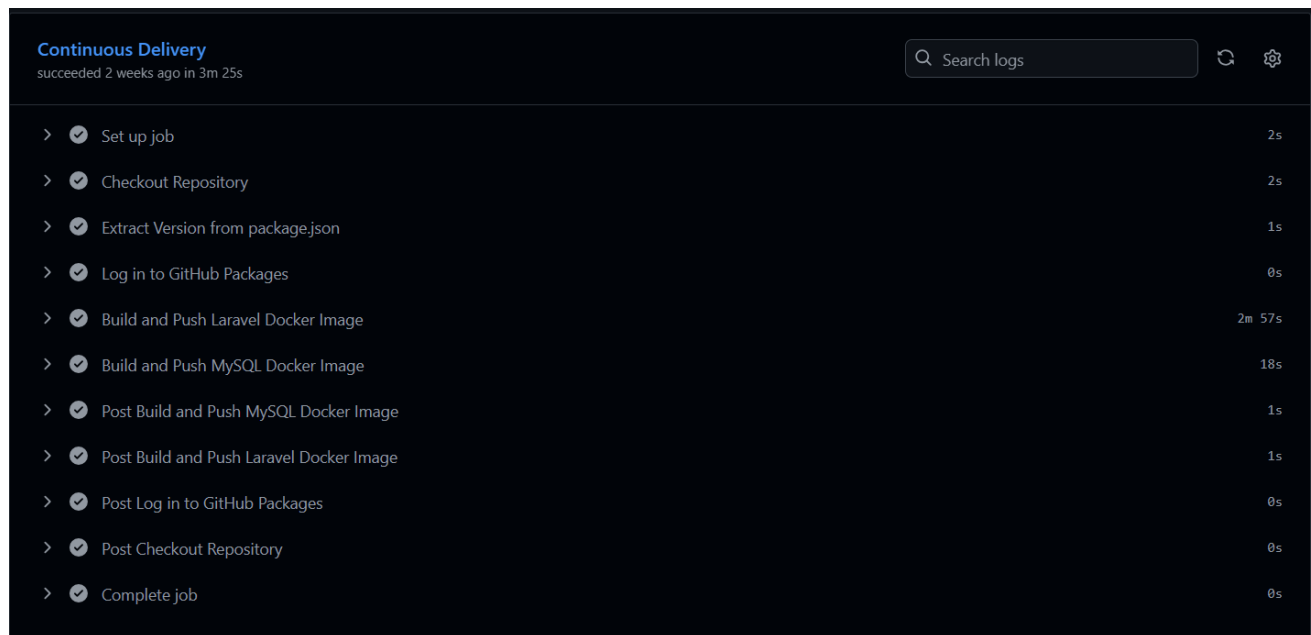
Visualisasi komparatif hasil penilaian terhadap sistem yang diberikan oleh empat pengguna teknis dan satu validator ahli ditampilkan pada Gambar 7.

Hasilnya menunjukkan bahwa pengguna teknis memberikan tingkat kelayakan sebesar 83,33%, sedangkan validator ahli memberikan nilai 85,71%. Keduanya termasuk dalam kategori sangat layak.

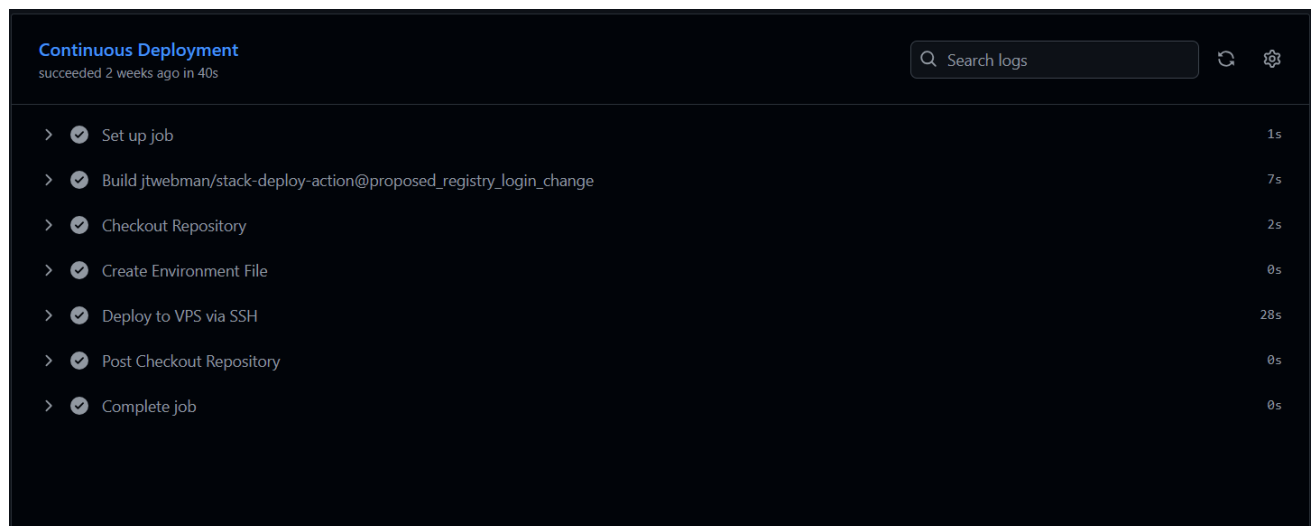
KESIMPULAN

Sistem otomasi deployment aplikasi berbasis web yang dibangun dengan metode Design Science Research (DSR) dan diterapkan menggunakan GitHub Actions serta Docker berhasil menyelesaikan permasalahan proses deployment yang sebelumnya dilakukan secara manual dan memerlukan waktu hingga dua jam. Sistem ini memungkinkan proses pengujian sampai pengiriman aplikasi ke server berjalan otomatis dan konsisten.

Berdasarkan pengujian blackbox terhadap tiga proses utama CI/CD, seluruh skenario pengujian berhasil dijalankan tanpa error, dengan tingkat akurasi sebesar 100%. Selain itu, pengujian terhadap waktu pelaksanaan menunjukkan bahwa rata-rata proses deployment hanya memerlukan waktu 4 menit 49 detik. Evaluasi melalui close beta testing terhadap empat pengguna teknis dan satu validator ahli menunjukkan tingkat kelayakan sebesar 83,33% dan



Gambar 4. Hasil Continuous Delivery



Gambar 5. Hasil Continuous Deployment

85,71%, yang termasuk kategori sangat layak.

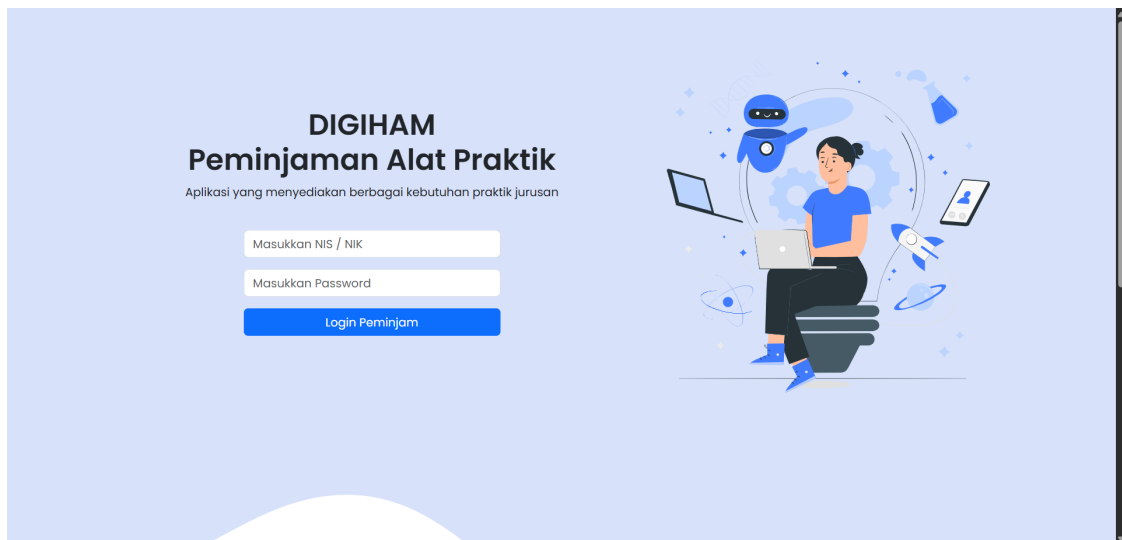
Hasil ini menunjukkan bahwa sistem otomasi deployment yang dirancang telah memenuhi aspek fungsionalitas, efisiensi, dan kemudahan penggunaan, serta dapat diimplementasikan dalam lingkungan SMK PGRI Wlingi.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada SMK PGRI Wlingi, khususnya kepada Bapak Oni Budi Setiyono selaku Kepala Jurusan RPL yang telah memberikan izin pelaksanaan penelitian, serta Ibu Nurul Fitrah Rahmadani atas dukungan dan bantuannya selama proses penelitian ini. Ucapan terima kasih juga disampaikan kepada para guru, teknisi, dan karyawan SMK PGRI Wlingi yang telah berpartisipasi dalam proses evaluasi sistem dan memberikan masukan yang bermanfaat bagi pengembangan penelitian.

DAFTAR PUSTAKA

- [1] J. Vom Brocke, A. Hevner, and A. Maedche, *Design Science Research Cases*, Progress in IS. Springer, 2020. [Online]. Available: <http://www.springer.com/series/10440>
- [2] J. Lord, *Building CI/CD Systems Using Tekton: Develop Flexible and Powerful CI/CD Pipelines Using Tekton Pipelines and Triggers*. Packt Publishing, Limited, 2021.
- [3] B. Laster, *Learning GitHub Actions: Automation and Integration of CI/CD with GitHub*. O'Reilly Media, Inc., 2023.
- [4] S. P. Kane and K. Matthias, *Docker: Up & Running*. O'Reilly Media, Inc., 2023. [Online]. Available: <http://oreilly.com>
- [5] A. B. Chaudhuri, *Flowchart and Algorithm Basics: The*



Gambar 6. Tampilan Web Digiham Sarana

Tabel 3. Pengujian Blackbox

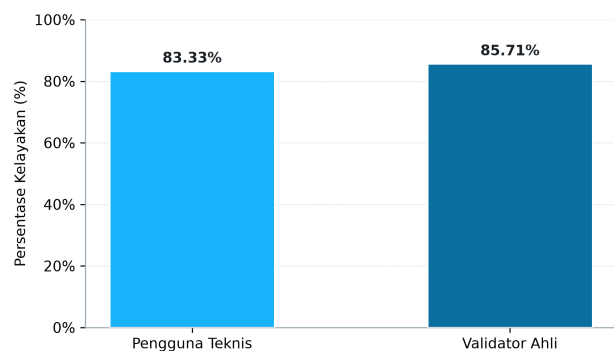
No	Proses	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Continuous Integration	Sistem berhasil memverifikasi kode dan menjalankan pengujian secara otomatis.	Pengujian otomatis berjalan lancar dan semua pengujian dinyatakan lulus.	Berhasil
2	Checkout Repository	Aplikasi berhasil dikemas dan dikirim ke layanan penyimpanan untuk siap digunakan.	Paket aplikasi berhasil dibuat dan dikirim tanpa kendala.	Berhasil
3	Continuous Deployment	Aplikasi berhasil dipasang di server dan dapat diakses melalui alamat yang tersedia.	Aplikasi berhasil dijalankan di server dan dapat diakses secara normal.	Berhasil

Tabel 4. Kategori Kelayakan

Persentase	Kategori Kelayakan
81% – 100%	Sangat Layak
61% – 80%	Layak
41% – 60%	Cukup Layak
21% – 40%	Tidak Layak
0% – 20%	Sangat Tidak Layak

Art of Programming. Mercury Learning and Information, 2020.

- [6] S. J. Putri, D. Galih, P. Putri, W. Hayuhardhika, and N. Putra, “Analisis Komparasi pada Teknik Black Box Testing (Studi Kasus: Website Lars),” *Journal of Internet and Software Engineering*, vol. 5, no. 1, 2024.
- [7] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 2015.
- [8] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R & D*. ALFABETA, CV., 2013.



Gambar 7. Diagram Batang Persentase Kelayakan