Implementasi dan Analisis Continuous Integration dan Continuous Deployment Berbasis GitHub Actions
pada Content Management System WordPress Dengan Shared Hosting
Kalep Firdaus, Ir. Yuris Mulya Saputra, S.T., M.Sc., Ph.D., IPM., SMIEEE
Universitas Gadjah Mada, 2026 | Diunduh dari http://etd.repository.ugm.ac.id/

INTISARI

IMPLEMENTASI DAN ANALISIS *CONTINUOUS INTEGRATION* DAN *CONTINUOUS DEPLOYMENT* BERBASIS GITHUB ACTIONS PADA *CONTENT MANAGEMENT SYSTEM* WORDPRESS DENGAN *SHARED HOSTING*

Kalep Firdaus

22/499648/SV/21352

Pengelolaan WordPress, sebagai *Content Management System* (CMS) yang menguasai 42,8% situs web di dunia, masih banyak dilakukan secara manual melalui protokol *File Transfer Protocol* (FTP), khususnya pada lingkungan *shared hosting* yang tidak menyediakan akses SSH. Metode ini berisiko menimbulkan *White Screen of Death* (WSOD) akibat ketiadaan mekanisme validasi otomatis dan kontrol versi. Penelitian ini bertujuan menerapkan dan menganalisis *pipeline Continuous Integration/Continuous Deployment* (CI/CD) menggunakan GitHub Actions pada WordPress di lingkungan *shared hosting*, serta membandingkannya dengan metode manual berbasis FileZilla FTP melalui pendekatan kuantitatif eksperimental. Pengujian dilakukan pada lima skenario yang mencakup *deployment time*, integritas direktori wp-content, deteksi kegagalan, waktu pemulihan (MTTR), dan ketahanan volume *file*. Hasil penelitian menunjukkan bahwa CI/CD unggul 56,64% pada *deployment* lima *file* dan 21,01% pada sepuluh *file*, meskipun manual lebih cepat pada perubahan satu *file* (39,06%) akibat *overhead* tetap *workflow*. CI/CD berhasil mendeteksi 66,67% kegagalan sebelum mencapai server, dibandingkan 0% pada manual, serta mempercepat MTTR sebesar 10,10% dengan prosedur yang deterministik dan jejak audit permanen. Pada pengujian volume 100 *file*, kedua metode menunjukkan keberhasilan setara (100%). Secara keseluruhan, penelitian ini membuktikan bahwa implementasi CI/CD menggunakan GitHub Actions memberikan peningkatan signifikan pada sebagian besar dimensi kerangka DORA dibandingkan metode manual, dengan keunggulan yang paling menonjol pada skenario perubahan berskala sedang hingga besar serta pemulihan layanan yang lebih andal dan minim risiko *human error*.

Kata kunci: CI/CD, GitHub Actions, WordPress, *shared hosting*, *deployment*

UNIVERSITAS
GADJAH MADA
**Implementasi dan Analisis Continuous Integration dan Continuous Deployment Berbasis GitHub Actions
pada Content Management System WordPress Dengan Shared Hosting**
Kalep Firdaus, Ir. Yuris Mulya Saputra, S.T., M.Sc., Ph.D., IPM., SMIEEE
Universitas Gadjah Mada, 2026 | Diunduh dari http://etd.repository.ugm.ac.id/

ABSTRACT

IMPLEMENTATION AND ANALYSIS OF CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT BASED ON GITHUB ACTIONS ON WORDPRESS CONTENT MANAGEMENT SYSTEM WITH SHARED HOSTING

Kalep Firdaus

22/499648/SV/21352

WordPress, as a Content Management System (CMS) that powers 42.8% of websites worldwide, is still largely managed manually via the File Transfer Protocol (FTP), particularly in shared hosting environments that do not provide SSH access. This manual approach carries a high risk of triggering a White Screen of Death (WSOD) due to the absence of automated validation and version control mechanisms. This study aims to implement and analyze a Continuous Integration/Continuous Deployment (CI/CD) pipeline using GitHub Actions for WordPress in a shared hosting environment and compare it with a conventional manual deployment method using FileZilla FTP through a quantitative experimental approach. Testing was conducted across five scenarios covering deployment time, wp-content directory integrity, failure detection, recovery time (MTTR), and file-volume resilience. The results show that CI/CD outperformed the manual method by 56.64% for five-file deployments and 21.01% for ten-file deployments, although the manual method was 39.06% faster for single-file changes due to fixed workflow overhead. CI/CD successfully detected 66.67% of injected failures before they reached the production server, compared to 0% for the manual method, and reduced MTTR by 10.10% through a deterministic recovery procedure with a permanent audit trail. In the 100-file volume resilience test, both methods achieved an equal 100% success rate. Overall, this study demonstrates that implementing CI/CD with GitHub Actions yields significant improvements across most dimensions of the DORA metrics framework compared to the manual method, with the most pronounced advantages observed in medium-to-large-scale change scenarios and more reliable, human-error-resistant service recovery.

Keywords: *CI/CD, GitHub Actions, WordPress, shared hosting, deployment*