

Penerapan *Continuous Integration* (CI) Pada Aplikasi Web Profil Karang Taruna (Studi Kasus : Karang Taruna Kabupaten Bengkalis)

Depandi Enda¹, Supria², Isna Yulia³, M. Farid Amirul⁴, M. Farel Asyrofi⁵
Politeknik Negeri Bengkalis^{1,2,3,4,5}
depandienda@polbeng.ac.id¹, phiya@polbeng.ac.id², isna@polbeng.ac.id³,
faridamirul268@gmail.com⁴, m.farelasyrofi@gmail.com⁵

Abstract

Along with the rapidly growing number of software products on the market, it has encouraged every development team to produce reliable and quality software and release updates as often as possible and on a scheduled basis. No doubt, making this software usually involves several people working in teams to produce software. However, there are problems in the code integration process that often cause new bugs when program code that several different developers have worked on is used as a single code base to be built and tested. Therefore, the implementation of continuous integration can be one of the solutions in the software development process carried out by the development team. This study proposes the application of continuous integration in the process of developing a youth profile web application. To validate the implementation results, trials were conducted on a clone, pull, push and deploy web applications released in the production environment. The test results show that all test activities have been carried out successfully, and any code changes or deploy actions have been successfully executed.

Keywords : *Continuous Integration, Software Development, Web Application, Profile App*

1. PENDAHULUAN

Dibeberapa tahun yang lalu, tim pengembang perangkat lunak seringkali bekerja secara sendiri-sendiri untuk menyelesaikan fitur yang telah diemban dengan memakan waktu dan proses yang cukup panjang (tidak dapat diprediksi), kemudian menggabung beberapa pekerjaan tersebut menjadi satu kesatuan. Proses penggabungan dilakukan jika hanya seluruh fitur telah selesai ditulis dan lolos uji secara mandiri oleh tim pengembang. Hal ini akan membuat pekerjaan menjadi rumit dan sulit ketika seluruh fitur yang telah selesai dikerjakan akan digabung menjadi satu basis kode, ini juga memungkinkan penemuan *bugs* baru ketika aplikasi di *build* tanpa adanya koreksi dalam waktu yang cukup lama (Hüttermann, 2012). Maka dalam penerapannya muncullah cara baru yang praktis dalam mengembangkan perangkat lunak menggunakan teknik pengembangan berkelanjutan yang terdiri dari *continuous integration* (CI), *continuous delivery* (CD), dan *continuous deployment* (CDE).

Continuous Integration (CI) adalah praktik pengembangan yang mapan dan secara luas telah digunakan dalam industri pengembangan perangkat lunak (Fitzgerald dan Stol, 2017), di mana anggota tim mengintegrasikan dan menggabungkan pekerjaan pengembangan (misalnya, kode) secara sering, misalnya beberapa kali per hari (Mohammad, 2016). Setiap integrasi diverifikasi oleh *build* otomatis (termasuk pengujian) untuk mendeteksi kesalahan integrasi secepat mungkin (Fowler, 2022). *Continuous Delivery* (CDE) ditujukan untuk memastikan aplikasi selalu dalam kondisi siap produksi setelah berhasil melewati pengujian otomatis dan pemeriksaan kualitas (Weber dkk., 2016; Humble, 2022). CDE menggunakan serangkaian praktik misalnya, CI, dan otomatisasi penyebaran untuk mengirimkan perangkat lunak secara otomatis ke lingkungan produksi. Sedangkan *Continuous Deployment* (CD) selangkah lebih maju yang secara otomatis dan terus-menerus menyebarkan aplikasi ke lingkungan produksi atau pelanggan (Weber dkk., 2016).

Istilah integrasi berkelanjutan (CI) merupakan bagian dari salah satu proses pengembangan pemrograman ekstrim (eXtreme Programming) yang berjumlah dua belas praktik. Dalam praktiknya CI menggunakan sistem kontrol kode sumber yang tersimpan di dalam sebuah repository tunggal. Sistem control kode sumber ini biasanya menggunakan aplikasi seperti Stash, Git/Github, BitBucket, Subversion, dan IBM Rational ClearCase. Beberapa penelitian yang telah membahas konsep dan praktik CI maupun CD telah dilakukan oleh (Toba dkk., 2022; Zhu dkk., 2016; Shahin dkk., 2017; Arachchi dan Perera, 2018; Noviantama dan W, 2021; Jaeni dkk., 2021;); Dimana berdasarkan kajian beberapa penelitian tersebut menunjukkan bahwa penerapan praktik CI dan CD mampu meningkatkan kinerja proses pengembangan perangkat lunak yang dimulai dari tahapan *build* dan *testing* hingga *deployment* ke lingkungan produksi.

Adapun tujuan penelitian ini yaitu menerapkan praktik *continuous integration* pada pengembangan aplikasi web profil karang taruna kabupaten Bengkalis. Dimana untuk menentukan apakah praktik CI sudah berhasil diterapkan dengan baik maka dibuatlah pengujian sederhana dengan melakukan *cloning*, *pull*, *push* artifak kode aplikasi web profil di perangkat lunak Version Control System Github. Tahapan *deployment* aplikasi juga diuji dari repositori kode proyek github ke *mirror repository* yang ada di CPANEL *web hosting*. Dengan adanya penerapan ini diharapkan dapat membantu tim pengembang dalam melakukan kontrol versi aplikasi web profil yang dibuat ketika aplikasi telah di *publish* di layanan web hosting atau telah berada di lingkungan produksi.

2. TINJAUAN PUSTAKA

Beberapa penelitian terdahulu yang membahas penerapan *continous integration* dalam pengembangan perangkat lunak yaitu Jaeni dkk dengan judul penelitian implementasi *continuous integration/continuous delivery (CI/CD)* pada *performance testing devops*. Permasalahan pengembangan dengan cara tradisional yaitu keterlambatan proses rilis aplikasi yang sering kali terjadi dan menyebabkan klien kecewa. Penerapan CI/CD dapat dijadikan solusi untuk menyelesaikan permasalahan tersebut, selain itu juga dapat menjadi jembatan antara tim pengembangan dan tim operasional. *Time-based metrics* digunakan untuk mengukur waktu yang dibutuhkan oleh proses CI/CD dan tradisional dalam melakukan seluruh tahapan yang dimulai dari *notifications*, *testing*, *build* dan *deployment* aplikasi. *Quality-based metrics* untuk mengukur kualitas dengan 2 aspek yaitu *test past rate* dan *number of bugs*. Dimana penerapan CI/CD mampu mempersingkat proses dan meningkatkan kinerja aplikasi jika dibandingkan dengan cara tradisional (Jaeni dkk., 2021).

Implementasi DevOps pada Pengembangan Aplikasi e-Skrining Covid-19 telah dilakukan oleh Tohirin dkk. Dimana masalah pada fase *build*, *test* dan *deploy* di dalam siklus hidup pengembangan sistem sering kali terjadi dan dapat memakan banyak waktu dan menyebabkan rilis tidak sesuai jadwal. DevOps telah digunakan pada pengembangan aplikasi e-Skrining Covid-19 berbasis Agile Scrum, untuk memberikan solusi otomatisasi pada fase *build*, *test* dan *deploy* menjadi lebih efisien. Hasil penelitian menunjukkan bahwa penerapan devops dapat mempermudah proses penggabungan kode antar pengembang, *build* harian menjadi lancar dan pemeriksaan kesehatan dan kelayakan kode terjadi pada setiap kali ada aktivitas *commit* dan *push* dari pengembang (Tohirin dkk., 2020).

Penelitian berikutnya implementasi *continuous integration* dan *continuous deployment* pada aplikasi *learning management system* di PT. Millennia Solusi Informatika dilakukan oleh Noviantama dan Wahyu W. Siklus pengembangan perangkat lunak yang digunakan adalah metode *scrum*. Dimana konsep CI/CD diterapkan untuk memenuhi alur pengembangan yang bersifat agile dan iterative. Hasil penelitian yang telah dilakukan menunjukkan penerapan konsep CI/CD memudahkan tim pengembang dan tim operasional bekerja secara praktis,

menghindari kesalahan yang terjadi oleh manusia dengan proses otomatisasi dan membantu dalam proses rilis aplikasi (Noviantama dan W, 2021).

Evaluasi metodologi CI/CD untuk pengembangan perangkat lunak dalam perkuliahan telah dilakukan oleh Toba dkk. Evaluasi dilakukan terhadap data yang dikumpulkan dalam sebuah sistem CI/CD menggunakan metode survey untuk menemukan informasi yang bermanfaat sebagai umpan balik terhadap sistem CI/CD dalam perkuliahan. Survei dilakukan pada masa ujian tengah semester hingga ujian akhir semester mahasiswa sarjana teknik informatika pada matakuliah rekayasa perangkat lunak. Hasil survey menunjukkan bahwa mayoritas mahasiswa sangat antusias dalam mendalami konsep CI/CD dalam metode pengembangan perangkat lunak yang mutakhir (Toba dkk., 2022).

Tidak hanya di area pengembangan aplikasi *web* dan *mobile* saja yang dapat menerapkan konsep praktis CI/CD, pada sistem tertanam dan IoT juga dapat diterapkan. Penelitian yang dilakukan oleh Aladina dkk dengan judul penerapan mekanisme *continuous deployment* dalam pengembangan dan pembaruan perangkat lunak sistem benam berbasis IoT. Penerapan CD dimaksudkan untuk mengatasi pembaruan perangkat lunak yang dilakukan secara manual dengan datang langsung ke lokasi dimana perangkat IoT tersebut berada yang dapat memakan waktu dan biaya. Pada implementasinya penelitian tersebut menggunakan Visual Studi Code sebagai kode editor, Github sebagai code repository, AWS sebagai layanan cloud, Jenkins sebagai kakas bantu untuk melakukan *continuous deployment* dan tiga buah ESP32 sebagai perangkat IoT. Hasil dari pengujian menunjukkan waktu rata-rata untuk melakukan pembaruan menggunakan praktik CD hanya membutuhkan waktu 63,5 detik (Aladina dkk., 2022).

Selanjutnya penelitian yang dilakukan oleh Nugraha dkk dengan judul penelitian penerapan DevOps pada sistem tertanam dengan ESP8266 menggunakan mekanisme Over The Air (OTA). Penelitian Nugraha mengusulkan sebuah infrastruktur DevOps untuk pengembangan sistem tertanam yang terdiri dari perangkat lunak PlatformIO, Github dan Travis CI. Infrastruktur ini digunakan untuk melakukan pembaruan sistem pada perangkat IoT ESP8266. Tahapan DevOps yang terdiri dari build dan test, release hingga deploy telah berhasil dilakukan secara otomatis. Dimana rata-rata waktu yang dibutuhkan untuk proses build dan test adalah sebesar 77,21 detik dan waktu deploy sebesar 1,41 detik. Dimana waktu untuk melakukan pembaruan pada sistem perangkat IoT berjalan lebih cepat jika dilakukan secara manual (NUGRAHA dkk., 2021).

Berdasarkan beberapa penelitian yang telah diuraikan dapat disimpulkan bahwa penerapan CI/CD memiliki beberapa keunggulan, seperti yang diungkapkan oleh Zhu dkk yaitu proses pengujian dapat dilakukan secara otomatis, mempercepat umpan balik dan mengurangi konflik *environment* antar tim pengembang dan tim operasi (Zhu dkk., 2016). Penerapan praktik berkelanjutan seperti *continuous integration*, *delivery* dan *deployment* juga memungkinkan sebuah organisasi dalam industri pengembangan perangkat lunak untuk sesering mungkin dalam merilis fitur baru yang handal dari sebuah produk (Shahin dkk., 2017). Selain itu juga dapat meningkatkan produktivitas pengembangan sebuah aplikasi, melalui penerapan rilis aplikasi dengan otomatisasi (Arachchi dan Perera, 2018).

Sehingga pada penelitian ini perlu diterapkan *continuous integration* dalam proses pengembangan aplikasi web profil karang taruna yang dibangun agar proses integrasi kode dari beberapa pengembang menjadi lebih mudah dan dapat menyederhanakan proses rilis fitur baru dari aplikasi menjadi lebih cepat dan efisien.

3. METODE PENELITIAN

Lokasi penelitian saat melakukan riset berada di Laboratorium *Software Development*, Jurusan Teknik Informatika, Politeknik Negeri Bengkalis.

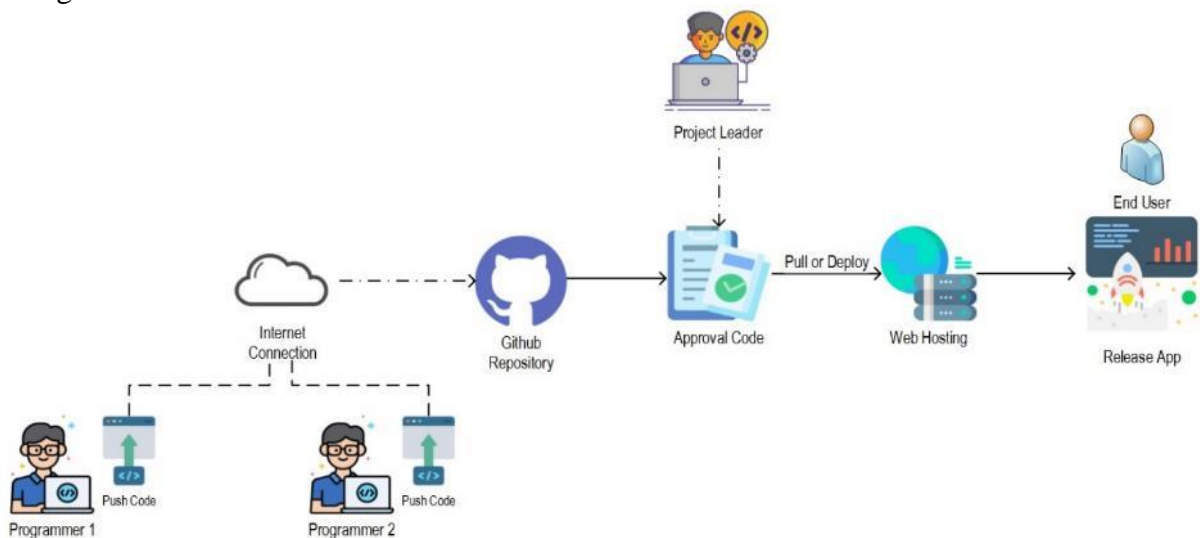
Perangkat keras yang digunakan untuk menerapkan CI/CD pada integrasi aplikasi web profil adalah sebagai berikut:

1. Laptop, digunakan untuk proses pengolahan data dengan spesifikasi sebagai berikut:
 - a. CPU : Intel Core i5 Gen. 8th
 - b. RAM : 16GB
 - c. Storage : SSD NVME 1 TB
 - d. Display Resolution : 1920x1080 (FHD)
2. Mouse Wireless, digunakan untuk mengenali input data yang dilakukan pengguna berkaitan dengan *event* klik, *drag* dan *on focus*.
3. Koneksi Internet yang stabil

Perangkat lunak yang dibutuhkan untuk penerapan CI/CD pada integrasi aplikasi web profil karang taruna adalah sebagai berikut :

1. Github web apps dan akunnya.
2. *Web Hosting CPANEL* yang memiliki fitur Git dan FTP.
3. Source Code Aplikasi Web Profil Karang Taruna.
4. Visual Studio Code sebagai IDE untuk pengembangan aplikasi.
5. *Web Browser* untuk menguji dan menjalankan aplikasi web profil.
6. XAMPP untuk menjalankan layanan Apache, Database MySQL dan PHP MyAdmin (dijalankan pada *localhost*).
7. Notepad++ (editor teks untuk dokumentasi proses penerapan)

Prosedur penerapan CI/CD pada integrasi aplikasi web profil karang taruna adalah sebagai berikut:



Gambar 1. Prosedur Deployment Aplikasi Web Profil Karang Taruna

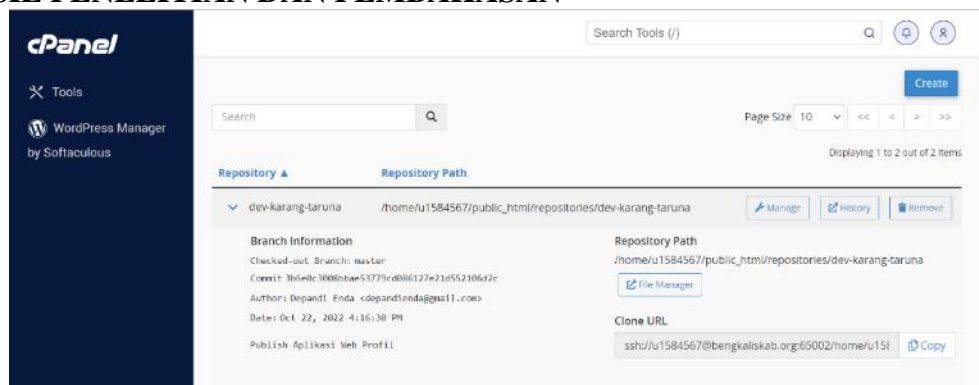
Pada Gambar 1, *project* aplikasi web profil karang taruna di kerjakan oleh dua orang *programmer* (1 & 2) dan seorang *project leader* sebagai koordinator dan penanggung jawab aktivitas *project* yang meliputi pekerjaan *review code*, *testing* dan *deploy* aplikasi ke *production*. Setiap *programmer* akan diberikan *task* berbeda yang telah disepakati. *Task* yang diberikan berupa bagian fitur aplikasi yang harus dipenuhi untuk mencapai *goal project*. Ketika suatu *task* yang diberikan telah selesai dikerjakan, maka *programmer* akan melakukan push kode kedalam *repositories* github yang telah disediakan oleh *project leader*. Kemudian *project leader* akan mengulas kode dan melakukan pengujian kode, jika pengujian berhasil dilakukan dan tidak terdapat permasalahan ketika dilakukan integrasi kedalam kode utama (*master*), maka *project leader* akan melakukan persetujuan atas kode tersebut dan akan menginisiasi dilakukannya perubahan *code* di web hosting. Proses inisiasi dilakukan melalui pada CPANEL web hosting. Dimana pada layanan Git Control Version yang ada di layanan web hosting dapat secara otomatis mengambil kode yang ada pada *repository github* untuk disalin ke *mirror repository web hosting*. Setelah proses update kode pada *main branch*

selesai dilakukan maka pengguna akhir dapat melihat perubahan aplikasi secara langsung sesaat ketika *project leader* telah melakukan persetujuan atau *push* kode ke *github*.

Adapun langkah untuk menerapkan CI dan CD pada aplikasi web profil adalah sebagai berikut:

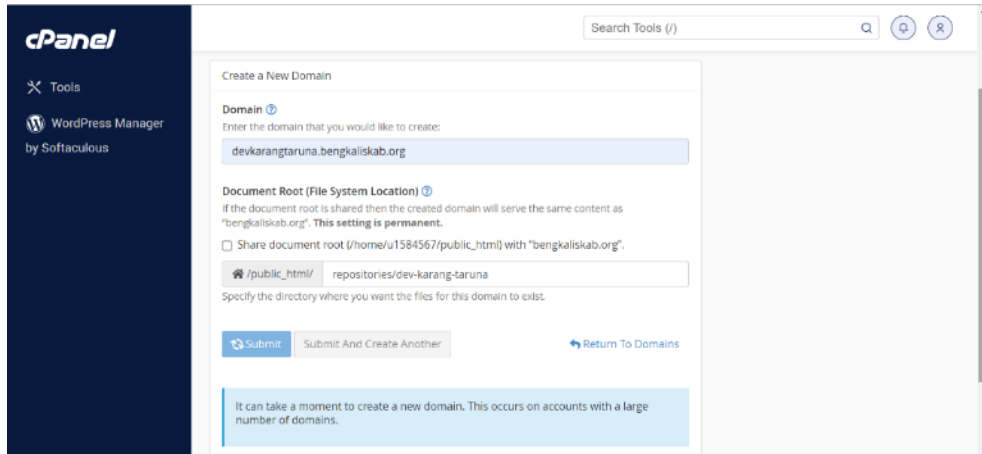
1. Pembuatan Repositori Aplikasi Web Profil dan Cloning ke CPANEL
 - a. Menghubungkan *Source Code* Lokal ke *Repositories*
 - b. Clone Repository di Git Version Control CPANEL
2. Konfigurasi Domain Web dan Database
 - a. Membuat Domain Baru di CPANEL
 - b. Konfigurasi Database di CPANEL
 - c. Membuat User Baru
 - d. Menambahkan User ke Database
 - e. Pengaturan Role User ke Database
 - f. Mengimpor Data ke Database
3. Ubah Pengaturan Database di *Source Code* lokal dan Push ke Github amati perubahan yang terjadi.
4. Menambahkan Programmer di Project Github Aplikasi Web Profil
5. *Cloning project* menggunakan Visual Studio Code IDE dan akun *Programmer 1*.
6. *Ubah* isi dari file README.md lakukan *commit* dan *push*, lalu amati perubahan di github.
7. Dengan cara yang sama lakukan *cloning project* menggunakan Visual Studio Code IDE, akan tetapi menggunakan akun github *Programmer 2*.
8. Lakukan perubahan terhadap file footer.php tambahkan teks “POLBENG” di baris copyright pada baris 215. Lakukan *commit* dan *push*, lalu amati perubahan yang terjadi di github.
9. Selanjutnya lakukan *deploy source code* yang telah ditambahkan oleh *Programmer 1* dan *Programmer 2* ke CPANEL. Amatilah perubahan pada halaman website bagian footer.

4. HASIL PENELITIAN DAN PEMBAHASAN



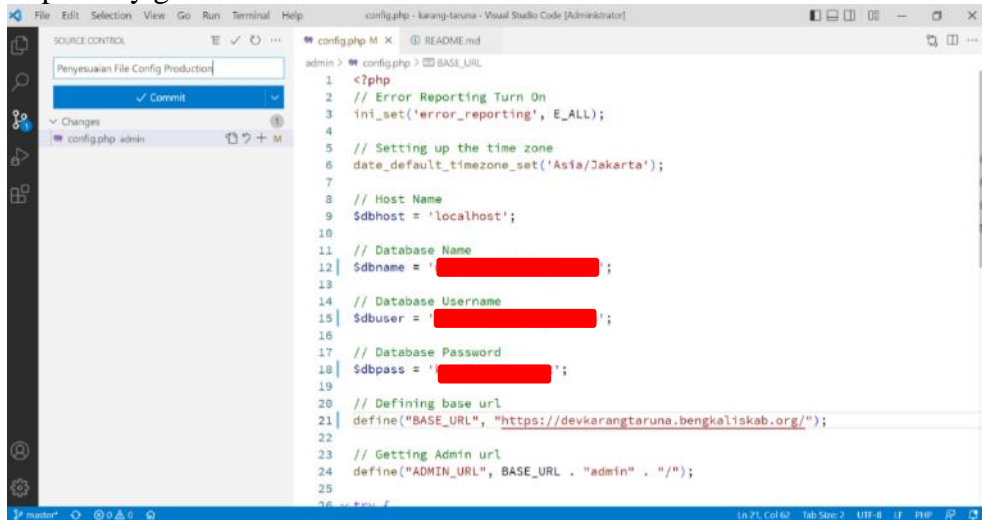
Gambar 2. Hasil Clone Repository dev-karang-taruna ke CPANEL

Hasil penyalinan *source code* aplikasi web profil dari *repository github* seperti pada Gambar 2. Dimana *branch* yang telah disalin adalah *branch master* dari *repository github*.

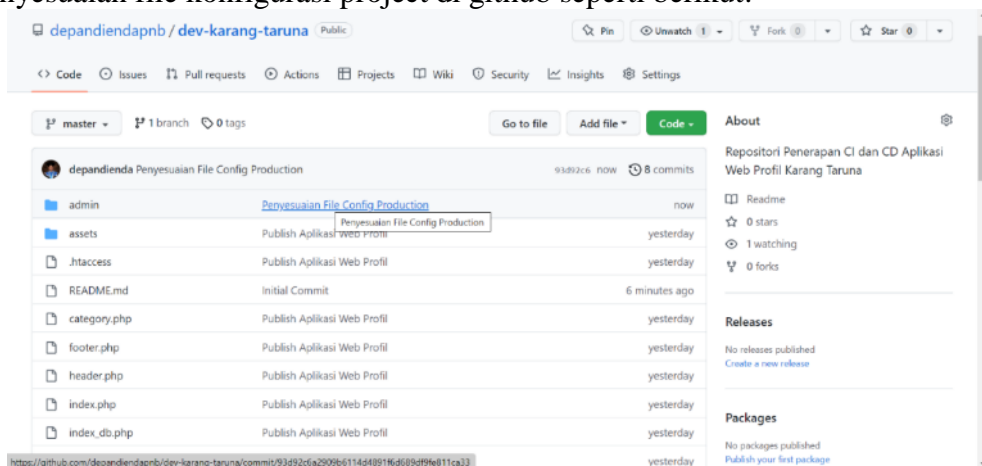


Gambar 3. Pembuatan Domain Baru

Domain baru dibuat dengan nama devkarangtaruna.bengkaliskab.org untuk keperluan uji coba. Dokumen root dari domain ini di arahkan ke repository mirror yang ada di CPANEL. Sehingga ketika terdapat perubahan di repository github maka repository mirror yang ada di CPANEL akan mendapat update secara berkala, mengikuti perubahan yang terjadi di repository github.

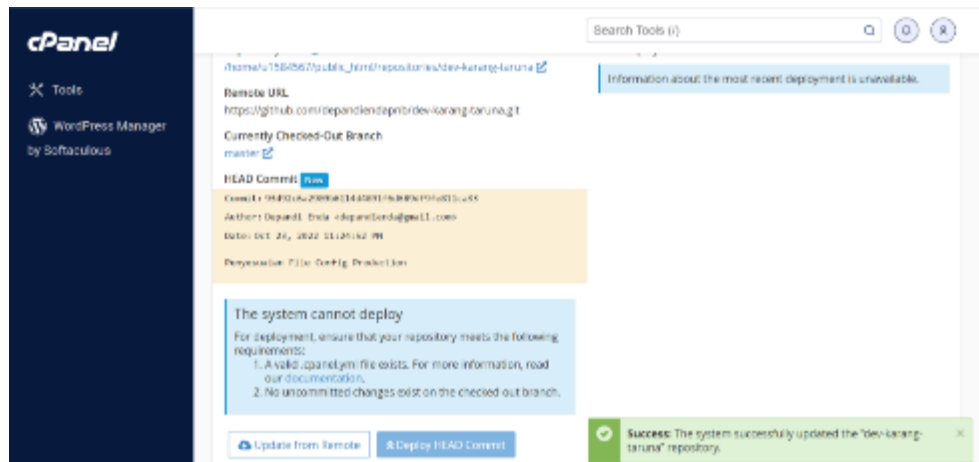


Gambar 4. Melakukan Perubahan File config.php, Commit dan Push ke Github Hasil penyesuaian file konfigurasi project di github seperti berikut:



Gambar 5. Hasil Perubahan File Konfigurasi Project di Github

Lakukan deploy setelah melakukan perubahan file konfigurasi project agar perubahan diterapkan di bagian production:



Gambar 6. Deploy perubahan file konfigurasi di CPANEL



Gambar 7. Hasil Deploy di Bagian Production

Selanjutnya uji coba penerapan CI dan CD dilakukan dengan aktivitas *clone*, *pull* dan *push* artifak file kode di github project, kemudian project leader melakukan pemeriksaan kode yang dipush oleh *programmer 1* dan *programmer 2* dan melakukan *deploy* hasil *push* kode ke CPANEL dan mengamati hasil perubahan kode di website yang telah di hosting. Hasil uji coba yang diperoleh adalah sebagai berikut:

Tabel 1. Hasil pengujian *clone*, *pull*, *push* dan *deploy* kode di Github Project

Developer	Clone Code	Pull Code	Push Code	Deploy by Project Leader
Project Leader	-	Success	Success	Success
Programmer 1	Success	Success	Success	-
Programmer 2	Success	Success	Success	-

Berdasarkan hasil pengujian *clone*, *pull*, dan *push* kode ke dalam *repository project* aplikasi web profil oleh *programmer 1* dan *programmer 2* berhasil dilakukan dan tercatat perubahannya ke dalam *repository*. Sedangkan *project leader* dapat melakukan *pull* dan *push project* yang ada direpositori github dengan sukses, *deploy project* juga berhasil dilakukan dari *repository github* ke dalam CPANEL, dan dapat dilihat perubahannya di halaman website yang telah ter-*publish*.

5. KESIMPULAN DAN SARAN

Penerapan CI adalah suatu hal yang sangat penting pada proses pengembangan aplikasi web profil yang dibuat. Hal ini dapat mempermudah tim pengembang dalam mengontrol versi perubahan perangkat lunak dan melakukan *deploy* ke bagian *production* ketika aplikasi sudah ter-*publish*. Tahapan CI pada aplikasi web profil karang taruna telah berhasil diterapkan dengan baik. Dimana setelah dilakukan pengujian aktivitas *clone*, *pull*, *push* dan *deploy* artifak kode oleh tim pengembang diperoleh hasil seluruh aktivitas telah

berhasil dilakukan dengan baik, dan setiap perubahan kode maupun aktivitas *deploy* sudah berhasil dijalankan.

6. DAFTAR PUSTAKA

- Aladina, Y. F., Bhawiyuga, A., Siregar, R. A., & ... (2022). Penerapan Mekanisme Continuous Deployment dalam Pengembangan dan Pembaruan Perangkat Lunak Sistem Benam Berbasis Internet of Things. ... *Informasi Dan Ilmu ...*, 9(3), 647–654. <https://doi.org/10.25126/jtiik.202295750>
- Arachchi, S. A. I. B. S., & Perera, I. (2018). Continuous integration and continuous delivery pipeline automation for agile software project management. *MERCon 2018 - 4th International Multidisciplinary Moratuwa Engineering Research Conference*, 156–161. <https://doi.org/10.1109/MERCon.2018.8421965>
- Fitzgerald, B., & Stol, K. J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123, 176–189. <https://doi.org/10.1016/j.jss.2015.06.063>
- Fowler, M. (2022). *Continuous Integration*. <https://martinfowler.com/articles/continuousIntegration.html>
- Humble, J. (2022). *Continuous Delivery vs Continuous Deployment*. <https://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/>
- Hüttermann, M. (2012). *DevOps for Developers*. Apress.
- Jaeni, S., N. A., & L., A. D. (2021). *IMPLEMENTASI CONTINUOUS INTEGRATION / CONTINUOUS DELIVERY (CI / CD) PADA PERFORMANCE TESTING DEVOPS* Abstraksi Keywords : Pendahuluan Tinjauan Pustaka Metode Penelitian. 2, 1–5.
- Mohammad, S. M. (2016). Continuous Integration and Automation. *IBM Journal of Research and Development*, 9(10), 1–45.
- Noviantama, I. G., & W, A. P. W. (2021). Implementasi Contionous Integration Dan Continous Deployment Pada Aplikasi Learning Management System Di Pt. Millennia Solusi Informatika. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 8(1), 183–186. <https://doi.org/10.33197/jitter.vol8.iss1.2021.744>
- NUGRAHA, A. W. W., ROSYADI, I., & KHOERULLATIF, F. (2021). Penerapan DevOps pada Sistem Tertanam dengan ESP8266 menggunakan Mekanisme Over The Air. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 9(3), 678. <https://doi.org/10.26760/elkomika.v9i3.678>
- Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5(Ci), 3909–3943. <https://doi.org/10.1109/ACCESS.2017.2685629>
- Toba, H., Gautama, T. K., Narabel, J., Widjaja, A., Sujadi, S. F., Suria, J., & No, S. (2022). *Evaluasi Metodologi CI / CD untuk Pengembangan*. 8(2), 227–234.
- Tohirin, T., Utami, S. F., Widiyanto, S. R., & Mauludyansah, W. Al. (2020). Implementasi DevOps Pada Pengembangan Aplikasi e-Skrining Covid-19. *Multinetics*, 6(1), 15–20. <https://doi.org/10.32722/multinetics.v6i1.2764>
- Weber, I., Nepal, S., & Zhu, L. (2016). Developing Dependable and Secure Cloud Applications. *IEEE Internet Computing*, 20(3), 74–79. <https://doi.org/10.1109/MIC.2016.67>
- Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). *DevOps and Its Practices* (pp. 32–34). IEEE COMPUTER SOCIETY.