

## **BAB VI**

### **PENUTUP**

#### **A. Kesimpulan**

Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan dari hasil analisis yang diperoleh sebagai berikut:

1. Proses otomatisasi proses pengembangan perangkat lunak menggunakan DroneCI dan ArgoCD berhasil dilakukan. Implementasi tersebut dapat digunakan untuk melakukan proses pengujian dalam integrasi *source code* dan *deployment* yang sebelumnya dilakukan manual oleh pengembang menjadi otomatis.
2. Implementasi yang diberikan dalam pengujian ini sepenuhnya dapat dilakukan dengan menggunakan perangkat sumber terbuka. Implementasi tersebut dapat dibangun menggunakan bantuan *tools* seperti ArgoCD, Terraform, DroneCI, MinIO dan Harbor.
3. Berdasarkan pengujian yang dilakukan dapat ditarik kesimpulan bahwa dengan menggunakan sistem tersebut pengembang dapat menghemat waktu secara bertahap dengan penghematan yang ditawarkan adalah 8 menit 37.8 detik untuk setiap proses *deployment* yang sebelumnya dilakukan secara manual. Penghematan bertumpuk tersebut dapat menghasilkan penghematan hingga 34.52 jam selama setahun. Perhitungan tersebut didasarkan dengan kegiatan *deployment* ke *environment staging* sebanyak 5 kali seminggu dan 1 kali seminggu untuk *environment production*.

#### **B. Saran**

Berdasarkan penelitian yang telah dilakukan, menurut kesimpulan yang diperoleh masih terdapat beberapa kekurangan. Oleh sebab itu memberikan beberapa saran yang dapat diterapkan agar kedepannya jika ada penelitian serupa dapat dilakukan dengan lebih baik. Saran yang dapat berikan antara lain:

1. Melakukan diversifikasi objek penelitian sehingga tidak hanya menjadikan satu objek sistem kompresi gambar melainkan sistem lainnya untuk mensimulasikan proses SDLC didalam perusahaan secara kompleks.
2. Implementasi yang dilakukan cukup kompleks untuk perusahaan kecil. Hal ini menjadikan implementasi yang dilakukan dicocokkan untuk perusahaan skala menengah-besar.



## DAFTAR PUSTAKA

- [1] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. New York, NY: McGraw-Hill Professional, 2014.
- [2] N. Forsgren, D. Smith, J. Humble, and J. Frazelle, "2019 Accelerate State of DevOps Report," 2019. [Online]. Available: <http://cloud.google.com/devops/state-of-devops/>
- [3] D. Smith, D. Villalba, M. Irvine, D. Stanke, and N. Harvey, "2021 Accelerate State of DevOps Report," 2021. [Online]. Available: <https://cloud.google.com/devops/state-of-devops/>
- [4] R. Ghimire, "Deploying Software in the Cloud with CI/CD Pipelines," Haaga-Helia University of Applied Sciences, Helsinki, 2020.
- [5] T. Tohirin, S. F. Utami, S. R. Widiyanto, and W. al Mauludyansah, "Implementasi DevOps Pada Pengembangan Aplikasi e-Skrining Covid-19," *MULTINETICS*, vol. 6, no. 1, pp. 15–20, May 2020, doi: 10.32722/multinetics.v6i1.2764.
- [6] S. Ferdian, T. Kandaga, A. Widjaja, H. Toba, R. Joshua, and J. Narabel, "Continuous Integration and Continuous Delivery Platform Development of Software Engineering and Software Project Management in Higher Education," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr. 2021, doi: 10.28932/jutisi.v7i1.3254.
- [7] A. Alperly and M. A. F. Ridha, "Implementasi CI/CD Dalam Pengembangan Aplikasi Web Menggunakan Docker dan Jenkins," *9th Applied Business and Engineering Conference*, vol. Vol. 9, pp. 287–296, 2021.
- [8] A. DHANY, "Implementation of Docker and Continuous Integration / Continuous Delivery for Management Information System Development," *IJEEIT International Journal of Electrical Engineering and Information Technology*, vol. 3, no. 2, pp. 20–24, Jan. 2021, doi: 10.29138/ijeeit.v3i2.1208.
- [9] M. Wittig and A. Wittig, *Amazon Web Services in Action*. 2015.
- [10] "Kubernetes (K8s)." <https://github.com/kubernetes/kubernetes> (accessed Feb. 02, 2022).
- [11] C. Doxsey, *Introducing Go: Build Reliable, Scalable Programs*, First Edition. Sebastopol, CA: O'Reilly Media, 2016.

- [12] “Drone CI/CD Documentation.” <https://docs.drone.io/> (accessed Feb. 02, 2022).
- [13] “Argo CD - Declarative Continuous Delivery for Kubernetes.” <https://github.com/argoproj/argo-cd/> (accessed Feb. 02, 2022).
- [14] “What is CI/CD?,” 2018. <https://www.redhat.com/en/topics/devops/what-is-ci-cd> (accessed Feb. 02, 2022).

