

RINGKASAN

PENGEMBANGAN APPLICATION PROGRAMMING INTERFACE (API) APLIKASI PENDETEKSI PENYAKIT TANAMAN MENGGUNAKAN METODE DEVOPS

Bagus Syafiq Faqihuddin

Penelitian ini mengkaji penerapan metodologi DevOps dalam pengembangan Application Programming Interface (API) untuk aplikasi pendeteksi penyakit tanaman berbasis kecerdasan buatan. DevOps dipilih sebagai pendekatan strategis untuk mengatasi tantangan dalam integrasi, *deployment*, dan *monitoring* API yang selama ini menjadi kendala utama dalam pengembangan backend yang efisien dan berkelanjutan. Implementasi mencakup empat pilar utama DevOps: *Continuous Development*, *Continuous Integration*, *Continuous Deployment*, dan *Continuous Monitoring*, dengan dukungan GitHub Actions sebagai sistem otomasi *pipeline* dan Docker pada *platform cloud* DigitalOcean.

Penelitian ini menghasilkan arsitektur modular *RESTful* API yang mencakup lima fitur utama: autentikasi, manajemen pengguna, prediksi penyakit tanaman, riwayat prediksi, dan fitur berbagi. *Backend* dikembangkan menggunakan Node.js dan Express, sedangkan pengolahan model dilakukan menggunakan Python dan Flask. Pengujian menunjukkan tingkat keberhasilan *deployment* sebesar 95,65% (22 dari 23), dengan waktu eksekusi *pipeline* antara 42 hingga 152 detik. *Monitoring* menunjukkan performa sistem yang stabil, meskipun ditemukan beberapa kebutuhan optimalisasi pada fitur manajemen pengguna dan riwayat. Hasil penelitian membuktikan bahwa pendekatan DevOps efektif diterapkan bahkan oleh pengembang individu dalam konteks proyek berskala kecil hingga menengah. Pendekatan ini meningkatkan kecepatan pengembangan, keandalan sistem, serta efisiensi kolaborasi antar tim pengembang.

Kata kunci : DevOps, CI/CD, DigitalOcean, GitHub Actions, RESTful API.

DEVELOPMENT OF AN APPLICATION PROGRAMMING INTERFACE (API) PLANT DISEASE DETECTION APPLICATION USING DEVOPS METHODOLOGY

Bagus Syafiq Faqihuddin

This study examines the application of the DevOps methodology in the development of an Application Programming Interface (API) for an artificial intelligence-based plant disease detection application. DevOps was chosen as a strategic approach to address challenges in the integration, deployment, and monitoring of APIs, which have long been major obstacles in the development of efficient and sustainable backends. The implementation includes four main pillars of DevOps: Continuous Development, Continuous Integration, Continuous Deployment, and Continuous Monitoring, supported by GitHub Actions as an automation pipeline system and Docker on the DigitalOcean cloud platform.

This study results in a modular RESTful API architecture that includes five main features: authentication, user management, plant disease prediction, prediction history, and sharing. The backend is developed using Node.js and Express, while model processing is done using Python and Flask. Testing showed a deployment success rate of 95.65% (22 out of 23), with pipeline execution times ranging from 42 to 152 seconds. Monitoring indicated stable system performance, although some optimization needs were identified in the user management and history features. The findings of this study prove that the DevOps approach is effective even for individual developers in the context of small to medium-sized projects. This approach enhances development speed, system reliability, and efficiency in collaboration among development teams.

Keywords: DevOps, CI/CD, DigitalOcean, GitHub Actions, RESTful API.