

## BAB 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

1. Penerapan **DevOps** dalam pengembangan **API** berhasil meningkatkan efisiensi dan keandalan sistem. Prinsip berkelanjutan memastikan **API** dapat diperbarui secara terus-menerus, menjaga kesesuaian antara pengembangan **API** dan kebutuhan aplikasi pendeteksi penyakit tanaman, serta meningkatkan respons terhadap perubahan yang diperlukan dalam pengembangan aplikasi.
2. Berdasarkan hasil *Continuous Integration* (CI) dan *Continuous Deployment* (CD), *pipeline* CI/CD terbukti efektif dalam memastikan stabilitas dan keandalan **API** di DigitalOcean. Pada CI, total 41 *run* dilakukan dengan 18 gagal dan 23 berhasil. Rata-rata waktu eksekusi untuk *run* yang berhasil adalah 93,35 detik, sementara untuk *run* yang gagal rata-rata waktu adalah 58,94 detik. Pada *Continuous Deployment* (CD), total 23 *deployment* dilakukan, dengan 22 berhasil (95,65%) dan 1 gagal yang kemudian berhasil setelah *retry*. Rata-rata waktu eksekusi *deployment* keseluruhan adalah 79,46 detik. Secara keseluruhan, *pipeline* CI/CD memastikan **API** dapat di-*deploy* secara efisien, mengurangi kesalahan, dan meningkatkan stabilitas **API**.
3. Berdasarkan hasil *monitoring* menggunakan k6 dan DigitalOcean Monitor, penerapan **DevOps** dalam pengembangan **API** menunjukkan bahwa fitur Autentikasi memiliki waktu respons yang lebih lambat, yaitu

812,71 ms, sedangkan fitur Prediksi mengalami keterlambatan signifikan dengan waktu respons rata-rata 3,5 detik. Di sisi lain, fitur Manajemen Pengguna, Riwayat, dan *Sharing* menunjukkan performa yang lebih cepat, dengan rata-rata waktu respons masing-masing 77,46 ms, 201,61 ms, dan 562,05 ms. Meskipun beberapa fitur memerlukan optimasi lebih lanjut, API tetap mampu beroperasi secara efisien dalam lingkungan produksi. Dari sisi performa server, penggunaan CPU maksimum tercatat mencapai 100% pada fitur Autentikasi dan Prediksi meskipun fungsionalnya berjalan sesuai dengan semestinya namun perlu optimasi, sedangkan fitur lain berada pada kisaran 43% hingga 75%, menunjukkan bahwa beban kerja masih dalam batas toleransi sistem. Secara keseluruhan, penerapan metode **DevOps** memberikan dampak positif dalam meningkatkan kualitas, stabilitas, dan efisiensi pengelolaan **API**.

## 5.2 Saran

1. Disarankan untuk mengoptimalkan *query* dengan indeks yang lebih efisien atau menggunakan *caching* untuk mengurangi beban akses *database*.
2. Disarankan untuk meningkatkan spesifikasi *server* atau menerapkan *horizontal scaling* dengan *load balancer* agar **API** dapat menangani lebih banyak *request* secara bersamaan.
3. Meskipun penelitian ini dilakukan secara *individu*, di masa depan disarankan untuk memperkuat kolaborasi antar tim dengan dokumentasi yang lebih lengkap dan jelas dalam setiap tahapan pengembangan **API**.