

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

1. Penelitian ini telah membuktikan bahwa *CI/CD pipeline* dapat diimplementasikan secara efektif pada arsitektur monolitik dan *microservices* dengan *auto-rollback* yang mampu mengatasi kegagalan *deployment* akibat kesalahan dependensi secara otomatis. Dengan demikian, *pipeline* ini andal dalam menjaga kestabilan sistem dan meningkatkan keandalan proses *deployment* pada kedua arsitektur tersebut.
2. Arsitektur *microservices* secara signifikan lebih efisien dalam menurunkan MTTR, mencerminkan kemampuan sistem untuk *recovery* lebih cepat dari kegagalan. MTTR pada *microservices* hanya berkisar 4,1–4,3 detik tanpa kesalahan dependensi, dan 26,35–28,8 detik saat terjadi kesalahan dengan waktu *rollback* 5,4–6,2 detik. Sebaliknya, arsitektur monolitik mencatat MTTR sebesar 19 detik tanpa kesalahan dan meningkat hingga 117,3 detik dengan *rollback* selama 26,85 detik ketika terjadi kesalahan dependensi.
3. Kesalahan dependensi berdampak signifikan terhadap kegagalan *deployment*, ditunjukkan dengan gagalnya seluruh versi terbaru berjalan di *production*. Hal ini menyebabkan MTTR meningkat drastis, dari 19 menjadi 117,3 detik pada arsitektur monolitik, dan dari 4,1–4,3 menjadi 26,35–28,8 detik pada arsitektur *microservices*. Meskipun keduanya terdampak, arsitektur *microservices* terbukti lebih tangguh dan efisien dalam menjaga keandalan sistem melalui penurunan MTTR yang lebih stabil.

5.2 Saran

1. Disarankan untuk melakukan migrasi penuh Moodle ke arsitektur *microservices* pada penelitian selanjutnya guna memperoleh gambaran yang lebih menyeluruh dan akurat mengenai performa sistem dalam kondisi operasional nyata. Namun, perlu juga dipertimbangkan metode atau arsitektur

lain yang dapat menangani skalabilitas secara efektif, sehingga tidak hanya terbatas pada migrasi ke *microservices*.

2. Penelitian lanjutan juga sebaiknya menggunakan jenis dependensi yang berbeda untuk mengevaluasi dampak variasi dependensi terhadap *downtime*, waktu *rollback*, dan keandalan *deployment* secara lebih komprehensif.
3. Selain MTTR, disarankan untuk menghitung metrik keandalan lain seperti *Mean Time Between Failures* (MTBF) untuk memahami frekuensi gangguan dan efektivitas arsitektur dalam mencegah kegagalan berulang.