

BAB V

KESIMPULAN

5.1 Kesimpulan

Pengembangan ini berhasil menerapkan arsitektur microservice untuk menggantikan sistem monolitik yang sebelumnya digunakan. Dalam implementasinya, sistem dipecah menjadi beberapa komponen layanan yang saling terpisah, terdiri dari satu frontend dan tiga backend yang masing-masing memiliki tanggung jawab terhadap proses tertentu. Pendekatan ini memberikan struktur yang lebih modular dan fleksibel. Ini memungkinkan proses development, test, dan deploy dari masing-masing layanan berjalan secara terisolasi dan independen.

Salah satu manfaat utama dari arsitektur microservice yang berhasil dicapai adalah peningkatan ketahanan sistem terhadap kegagalan. Jika salah satu layanan mengalami kendala atau tidak aktif, layanan lainnya tetap dapat beroperasi dengan normal tanpa terpengaruh. Hal ini memungkinkan sistem untuk tetap melayani permintaan pengguna pada proses-proses yang tidak terdampak, sehingga meminimalisir downtime total dan meningkatkan ketersediaan serta keandalan layanan secara keseluruhan.

Dengan hasil ini, dapat disimpulkan bahwa pendekatan microservice efektif dalam meningkatkan skalabilitas dan daya tahan sistem terhadap kegagalan.

5.2 Saran

Untuk pengembangan sistem ke depannya, disarankan agar diintegrasikan fitur satu sehat untuk mengintegrasikan rekam medis pasien secara real dan mengembangkan fitur yang mungkin belum sempurna.

Selain itu diharapkan arsitektur microservice dilengkapi dengan observability tools seperti Datadog atau Grafana. Penggunaan tools ini dapat membantu dalam memantau performa masing-masing layanan (*service*) dan mengawasi lalu lintas permintaan (*request*) yang masuk,

sehingga proses analisis kesalahan (*debugging*) dan optimalisasi sistem dapat dilakukan secara lebih efisien.

Selain itu, proses *deployment* aplikasi dapat ditingkatkan dengan memanfaatkan Kubernetes sebagai platform orkestrasi. Dengan menggunakan pods container yang terkelola secara otomatis, sistem dapat dimonitor secara *real-time*, diatur skalabilitasnya, dan memiliki ketahanan yang lebih baik terhadap gangguan layanan.

Disarankan pula untuk mengimplementasikan CI/CD (Continuous Integration/Continuous Deployment) guna mempercepat siklus pengembangan, mengurangi kesalahan manual dalam proses deployment, serta memastikan setiap perubahan kode dapat diuji dan dideploy dengan lebih cepat dan konsisten.