

BAB IV

KESIMPULAN

4.1 Kesimpulan

Pengembangan aplikasi Kalana Commerce telah berhasil diselesaikan dengan mengintegrasikan berbagai teknologi modern guna menciptakan platform e-commerce yang tangguh dan cerdas. Seluruh komponen utama, mulai dari infrastruktur backend berbasis Node.js dan database PostgreSQL yang dikelola melalui Prisma ORM, hingga sistem pembayaran otomatis menggunakan Midtrans, telah terbukti berjalan secara sinkron dan stabil. Keberhasilan sistem ini juga didukung oleh pengamanan jalur komunikasi yang berlapis menggunakan Cloudflare dan Nginx, serta implementasi autentikasi JWT dan enkripsi Bcrypt yang efektif melindungi data pengguna dari akses tidak sah. Validasi teknis terhadap seluruh fungsionalitas ini telah teruji secara nyata melalui pencapaian sebagai Juara 1 pada ajang Hackathon, yang menegaskan bahwa arsitektur yang dibangun telah memenuhi standar kualitas industri.

Selain aspek fungsionalitas dasar, aplikasi ini juga memberikan nilai inovasi melalui fitur Chatbot AI yang berperan sebagai asisten belanja adaptif. Keberhasilan implementasi fitur cerdas ini, yang dipadukan dengan efisiensi kontainerisasi melalui Docker, memastikan bahwa aplikasi memiliki skalabilitas yang baik serta kemudahan dalam proses deployment di lingkungan produksi. Secara keseluruhan, proyek ini tidak hanya mencapai target teknis yang ditetapkan, tetapi juga memberikan solusi belanja yang modern dan efisien, menjadikannya sebuah platform yang andal dan siap untuk dikembangkan lebih lanjut di masa mendatang.

4.2 Saran

Berdasarkan hasil pengembangan dan evaluasi yang telah dilakukan terhadap aplikasi Kalana Commerce, penulis menyadari bahwa masih terdapat ruang untuk pengembangan lebih lanjut guna meningkatkan performa dan jangkauan layanan. Oleh karena itu, penulis merumuskan beberapa saran strategis sebagai berikut:

a. Optimasi Performa melalui Strategi Caching dan Database

Untuk meningkatkan kecepatan respon API dalam menangani trafik yang lebih besar di masa mendatang, disarankan untuk mengimplementasikan sistem caching pada sisi server menggunakan Redis. Mekanisme ini akan sangat efektif untuk menyimpan data katalog produk yang jarang berubah, sehingga mengurangi beban kueri langsung ke database PostgreSQL dan mempercepat distribusi informasi ke aplikasi mobile. Selain itu, penelitian selanjutnya dapat mengeksplorasi penggunaan database indexing yang lebih mendalam pada Prisma ORM untuk mempercepat proses pencarian produk berdasarkan lokasi atau kategori tertentu.

b. Peningkatan Skalabilitas dengan Arsitektur Cloud-Native

Meskipun penggunaan Docker Compose saat ini telah menjamin stabilitas dalam proses deployment, pengembangan selanjutnya dapat diarahkan pada penggunaan teknologi orkestrasi kontainer seperti Kubernetes atau Docker Swarm. Implementasi ini memungkinkan sistem untuk melakukan Auto-scaling secara otomatis, yaitu menambah atau mengurangi jumlah kontainer berdasarkan beban CPU dan memori secara real-time. Hal ini penting untuk memastikan aplikasi tetap responsif meskipun terjadi lonjakan pengguna secara drastis pada waktu-waktu tertentu.

c. Pengembangan Model AI Lokal dan Personalisasi Mendalam

Integrasi AI saat ini masih sangat bergantung pada ketersediaan API pihak ketiga (LLM API). Untuk pengembangan ke depan, penulis menyarankan eksplorasi penggunaan model bahasa kecil (Small Language Model) yang dijalankan secara lokal di server internal menggunakan teknologi seperti Ollama atau vLLM di dalam kontainer Docker. Hal ini tidak hanya akan menekan biaya operasional API, tetapi juga meningkatkan privasi data pengguna dan memungkinkan personalisasi rekomendasi produk yang lebih spesifik berdasarkan riwayat belanja pelanggan tanpa data harus keluar dari infrastruktur perusahaan.