

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

- 1. Implementasi Domain-Driven Design (DDD) pada DANA Shop**
Penerapan DDD dalam sistem DANA Shop memberikan struktur yang lebih modular dan terorganisir. Dengan membagi domain bisnis menjadi beberapa bagian yang lebih spesifik, pengelolaan dan pengembangan sistem menjadi lebih mudah. Hal ini juga membantu meningkatkan skalabilitas sistem, sehingga setiap komponen dapat dikembangkan tanpa berdampak besar pada keseluruhan sistem.
- 2. Pengaruh Caching terhadap Performa Sistem**
Penggunaan Redis sebagai caching mechanism berhasil mengurangi waktu akses data dan beban pada database utama. Dengan adanya caching, permintaan data yang sering digunakan dapat dilayani lebih cepat tanpa harus selalu mengakses database PostgreSQL. Hasil pengujian menunjukkan bahwa caching memberikan peningkatan signifikan dalam waktu respon API dan throughput sistem.
- 3. Peningkatan Performa Sistem**
Implementasi DDD dan caching secara bersamaan memberikan dampak positif terhadap performa DANA Shop. Waktu respon API mengalami penurunan yang signifikan, throughput meningkat, serta efisiensi penggunaan sumber daya lebih optimal. Dibandingkan dengan platform social commerce lain seperti TikTok Shop dan Instagram Shop, DANA Shop menunjukkan performa yang lebih stabil dengan error rate yang lebih rendah.
- 4. Efisiensi Penggunaan Sumber Daya**
Dengan adanya optimasi melalui DDD dan caching, konsumsi CPU dan memori pada server mengalami penurunan yang berarti. Hal ini menunjukkan bahwa sistem dapat bekerja lebih efisien dalam menangani jumlah permintaan yang lebih tinggi tanpa meningkatkan konsumsi sumber daya secara signifikan.

5.2 Saran

1. **Pengujian dengan Data Real-Time**

Penelitian ini dilakukan dengan menggunakan data simulasi. Untuk mendapatkan hasil yang lebih akurat, pengujian selanjutnya dapat dilakukan dengan data real-time dari pengguna aktif guna mengukur efektivitas implementasi DDD dan caching dalam kondisi yang lebih mendekati produksi.

2. **Optimasi Caching Lebih Lanjut**

Studi ini menggunakan Redis sebagai caching utama. Untuk meningkatkan performa lebih lanjut, dapat dipertimbangkan penerapan teknik caching tambahan seperti edge caching atau content delivery network (CDN) agar sistem lebih responsif dalam menangani permintaan dari berbagai lokasi geografis.

3. **Evaluasi dalam Skala Beban yang Lebih Besar**

Pengujian performa dilakukan dalam skala yang terbatas. Untuk memastikan sistem dapat menangani beban yang lebih besar, sebaiknya dilakukan stress testing dan load testing dalam lingkungan dengan pengguna yang lebih banyak secara bersamaan.

4. **Integrasi dengan Teknologi Lain**

Mengingat perkembangan social commerce yang pesat, pengembangan sistem ke depan dapat mempertimbangkan integrasi dengan teknologi seperti machine learning untuk personalisasi rekomendasi produk atau implementasi event-driven architecture untuk meningkatkan skalabilitas sistem.

