

BAB V PENUTUP

5.1. Kesimpulan

Pada bagian ini, akan dijelaskan kesimpulan yang didapatkan berdasarkan tujuan penelitian yang telah dirumuskan sebelumnya dan hasil analisis yang telah diperoleh pada Bab IV

1. Dampak *Race Condition* dan *Efektivitas Message Queuing*:

Hasil pengujian menunjukkan bahwa sistem *baseline* mengalami *race condition* yang menyebabkan inkonsistensi data antara *stock quantity* dan *successful order* ditandai dengan *mismatch rate* lebih dari 0%. Sedangkan sistem *broker-based* mampu mengatasi *mismatch rate* hingga 0% pada seluruh skenario kombinasi pengujian pada penelitian ini yang menandakan keefektifan hingga 100% dalam mengatasi *race condition*, meningkatkan *throughput*, mengurangi *duration*, dan menghasilkan *latency* yang relatif rendah. Berdasarkan hasil ini, sistem *broker-based* memiliki performa yang lebih unggul dibandingkan sistem *baseline*.

2. Kemampuan Menangani Beban Tinggi dan Efisiensi pada Skala Besar:

Sistem *broker-based* terbukti berhasil dalam menangani kombinasi *requests* dan tingkat *concurrency* tinggi. Hal ini menunjukkan bahwa penggunaan RabbitMQ sebagai *message broker* sangat efektif dalam menangani risiko *race condition* yang sering terjadi pada sistem tanpa mekanisme pengendalian yang baik. Pada skenario pengujian dengan jumlah *requests* dan *concurrency* tinggi, sistem *broker-based* tetap mempertahankan performa yang stabil. Hal ini membuktikan bahwa solusi ini cocok untuk diterapkan pada sistem yang membutuhkan skalabilitas tinggi.

5.2. Saran

Bagian ini berisi rekomendasi untuk penelitian atau pengembangan lebih lanjut berdasarkan temuan dalam penelitian ini.

1. Penggunaan Perangkat Keras dan Sistem yang Lebih Baik:

Untuk meningkatkan performa dan memastikan hasil yang lebih optimal, disarankan menggunakan perangkat dengan spesifikasi lebih tinggi dan sistem yang lebih baik. Hal ini akan memungkinkan sistem *broker-based* untuk menangani *throughput* yang lebih besar dan mengurangi *latency* lebih lanjut.

2. Pengujian pada Kondisi Nyata:

Penelitian selanjutnya dapat mengimplementasikan sistem ini pada skenario dunia nyata, seperti aplikasi *e-commerce* atau sistem reservasi tiket, untuk menguji keandalannya dalam kondisi penggunaan sebenarnya.

3. Optimasi Konfigurasi RabbitMQ:

Penelitian lebih lanjut dapat difokuskan pada optimasi konfigurasi RabbitMQ, seperti pengaturan *queue*, *prefetch count*, dan *persistence*, untuk meningkatkan efisiensi sistem *broker-based*.

4. Penggunaan Message Broker Alternatif:

Selain RabbitMQ, disarankan untuk mengeksplorasi *message broker* lain seperti *Kafka* atau *NATS* untuk membandingkan performa dan efektivitas dalam mitigasi *race condition*.

5. Studi Tentang Overhead Broker-Based System:

Meskipun *broker-based system* memiliki keunggulan dalam mengatasi *race condition*, *overhead* yang dihasilkan oleh *broker* dapat menjadi topik menarik untuk diteliti lebih lanjut guna mengetahui bagaimana hal ini memengaruhi performa sistem secara keseluruhan.