OPTIMALISASI SKALABILITAS (STUDI KASUS: E-COMMERCE)

SKRIPSI

Rifqi Ramdhani

20210040037

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS TEKNIK, KOMPUTER DAN DESAIN

UNIVERSITAS NUSA PUTRA

SUKABUMI

2025

OPTIMALISASI SKALABILITAS E-COMMERCE

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat Dalam Menempuh

Gelar Sarjana Teknik Informatika

Rifqi Ramdhani

20210040037

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK, KOMPUTER DAN DESAIN

UNIVERSITAS NUSA PUTRA

SUKABUMI

2025

PERNYATAAN PENULIS

JUDUL :PENERAPAN METODE EVENT DRIVEN DESIGN (EDD) MENGGUNAKAN *RABBITMQ* DAN *DOCKER* UNTUK OPTIMALISASI SKALABILITAS (STUDI KASUS: E-COMMERCE)

NAMA : RIFQI RAMDHANI

NIM 20210040037

"Saya Menyatakan dan bertanggung jawab dengan sebenarnya bahwa Skripsi ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan masing-masing telah saya jelaskan sumbernya. Jika ada pihak lain yang mengklaim bahwa Skripsi ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar Sarjana Komputer/Sarjana Teknik saya beserta segala hak dan kewajiban yang melekat pada gelar tersebut"

Nusa Putra UNIVERSITY 2002 AMOR DEUS, PARENTIUM, CONCERVIS

Sukabumi, 14 Juli 2025

RIFQI RAMDHANI

Penulis

PENGESAHAN SKRIPSI

JUDUL :PENERAPAN METODE EVENT DRIVEN DESIGN MENGGUNAKAN *RABBITMQ* DAN *DOCKER* UNTUK OPTIMALISASI SKALABILITAS (STUDI KASUS: E-COMMERCE)

NAMA : Rifqi Ramdhani

NIM 20210040037

Skripsi ini telah diujikan dan dipertahankan di depan Dewan penguji pada sidang skripsi tanggal 14 Juli 2025. Menurut pandangan kami, Skripsi ini memadai dari segi kualitas untuk tujuan penganugerahan gelar sarjana computer (S.Kom).

Sukabumi, 14 Juli 2025

Pembimbing I	Pembimbing II
Allun Sujaada, S.Kom, M.T	Nugraha, M.Kom
NIDN.0718108001	NIDN.0413098904
Ketua Penguji	Ketua Program Studi

Angun Fergina, M.Kom	Somantri, ST., M.Kom
NIDN. 0407029301	NIDN.0419128801

Dekan Fakultas Teknik Komputer dan Desain

Ir. Paikun, S.T., M.T., IPM., ASEAN Eng

NIDN.0402037401

HALAMAN PERUNTUKAN

Alhamdulilah puji dan syukur saya panjatkan kepada Allah SWT, yang telah memberikan rahmat dan karunianya kepada setiap hambanya. Terima kasih atas setiap kemudahan dan rasa kesabaran yang telah diberikan saya sehingga karya sederhana ini dapat diselesaikan dengan tepat waktu.

Karya sederhana ini ku persembahkan untuk kedua orang tua, kakak, adek maupun keluarga besar yang telah memberikan suport selama 4 tahun ini. Terimakasih untuk setiap doa dan suportnya selama ini, semoga karya yang sederhana ini menjadi pijakan untuk gerbang kesuksesan anakmu ini nantinya.

karya sederhana ini ku persembahkan kepada diriku yang sudah berjuang selama 4 tahun ini. Terimakasih atas setiap kesabarannya untuk bisa menyelesaian tugas akhir ini, semoga ini menjadi awal dari setiap kesuksesan yang akan digapai kelak.

Nusa Putra
UNIVERSITY
2002
AMOR DEUS, PARENTIUM, CONCERVIS

ABSTRAK

Seiring dengan pesatnya pertumbuhan aplikasi e-commerce, tantangan utama yang dihadapi adalah menangani lonjakan jumlah *request* dan volume data, terutama saat puncak transaksi. Berdasarkan data perilaku pengguna e-commerce dari Kaggle (April 2020), aktivitas *Add to Cart* mencapai puncaknya pada pukul 09.00 dengan 6.658 aktivitas per jam, mencerminkan beban sistem yang tinggi dalam waktu singkat. Aplikasi e-commerce harus mampu menangani interaksi dan transaksi pengguna secara efisien serta mendukung skalabilitas. Untuk itu, pendekatan *Event Driven Architecture (EDA)* hadir sebagai solusi, dengan memanfaatkan *message broker* seperti RabbitMQ untuk mengelola komunikasi antar layanan secara asinkron.

Penelitian ini mengembangkan prototipe aplikasi e-commerce menggunakan RabbitMQ sebagai *message broker* dan Docker untuk pengelolaan layanan berbasis event. Fokus utama penelitian adalah mengevaluasi efektivitas arsitektur ini dalam menangani *request* berjumlah besar selama proses penambahan produk ke keranjang. Pengujian dilakukan melalui metode *Load Testing* menggunakan K6 untuk mengukur performa berdasarkan metrik latensi, throughput, dan skalabilitas.

Hasil pengujian menunjukkan bahwa sistem dengan pendekatan Event-Driven Design (EDD) mampu mempertahankan performa yang stabil bahkan saat jumlah pengguna meningkat. Pada pengujian dengan 50 Virtual Users selama 30 detik, sistem EDD berhasil menangani hingga 14.859 permintaan dengan response time rata-rata 49,96 ms dan throughput sebesar 493,82 request/s. Sebaliknya, sistem dengan pendekatan Remote Procedure Call (RPC) pada skenario yang sama hanya mampu menangani 169 permintaan dengan rata-rata response time sebesar 56,90 ms dan throughput sebesar 4,44 request/s. Perbedaan ini menunjukkan bahwa penerapan RabbitMQ sebagai message broker yang didukung oleh Docker dapat meningkatkan skalabilitas dan performa sistem secara signifikan.

Kata Kunci: *RabbitMQ*, *Docker*

ABSTRACT

As e-commerce applications continue to grow rapidly, a major challenge is handling the increasing number of requests and data volume, especially during peak transaction periods. Based on e-commerce behavior data from Kaggle (April 2020), *Add to Cart* activity peaked at 9:00 AM with 6,658 actions per hour, indicating a significant traffic surge in a short time. E-commerce platforms must efficiently manage user interactions and transactions while supporting high scalability. To address this, the *Event Driven Architecture (EDA)* offers an effective solution, where system changes are represented by events communicated asynchronously through a message broker such as RabbitMQ.

This research developed an e-commerce prototype implementing RabbitMQ as the message broker and Docker for managing event-based services. The study aims to evaluate the effectiveness of this architecture in handling high request volumes during the add-to-cart process. Load testing using K6 was conducted to assess system performance based on metrics such as latency, throughput, and scalability.

The test results show that the system using the Event-Driven Design (EDD) approach is able to maintain stable performance even as the number of users increases. In the test scenario with 50 Virtual Users over 30 seconds, the EDD-based system successfully handled up to 14,859 requests with an average response time of 49.96 ms and a throughput of 493.82 requests per second. In contrast, the system using the Remote Procedure Call (RPC) approach under the same conditions was only able to handle 169 requests, with an average response time of 56.90 ms and a throughput of 4.44 requests per second. This difference indicates that the implementation of RabbitMQ as a message broker, supported by Docker, can significantly enhance the scalability and performance of the system.

Keywords: *RabbitMQ*, *Docker* Event Driven Design

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT, berkat rahmat dan karunia-Nya akhirnya penulis dapat menyelesaikan skripsi sebagai salah satu syarat untuk menyelesaikan pendidikan di Universitas Nusa Putra Sukabumi. Tujuan penulisan skripsi ini adalah untuk memberikan kontribusi terhadap ilmu pengetahuan dan praktik di bidang teknik informatika. Sehubungan dengan itu penulis menyampaikan penghargaan dan ucapan terima kasih yang sebesar-besarnya kepada:

1. Orang tua dan keluarga yang selalu memberikan doa, dukungan moral dan material kepada penulis.
2. Bapak Dr. Kurniawan ST, M.Si, MM Rektor Universitas Nusa Putra Sukabumi
3. Bapak Ir. Paikun, S.T., M.T., IPM., ASEAN Eng Ketua Fakultas Teknik, Komputer dan Desain Universitas Nusa Putra Sukabumi.
4. Bapak Ir. Somantri., S.T., M. Kom Ketua Program Studi Universitas Nusa Putra Sukabumi.
5. Bapak Alun Sujjada, S.Kom, M.T selaku Dosen Pembimbing I Universitas Nusa Putra Sukabumi.
6. Bapak Nugraha, M.Kom selaku Dosen Pembimbing II Universitas Nusa Putra Sukabumi.
7. Para Dosen Program Studi Teknik Informatika Universitas Nusa Putra Sukabumi, atas ilmu, motivasi, dan dorongan yang telah diberikan kepada penulis selama masa perkuliahan.
8. Rekan –rekan mahasiswa di Program Studi Teknik Informatika Universitas Nusa Putra Sukabumi, yang telah memberikan dukungan, bantuan dan semangat selama proses penulisan skripsi ini
9. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu dan mendukung penulis selama proses penyelesaian skripsi ini.

Penulis menyadari bahwa penelitian ini tidak akan terwujud tanpa adanya dukungan dari keluarga, teman-teman yang telah memberikan semangat moral sepanjang

perjalanan ini. Terimakasih penulis sampaikan kepada semua orang yang telah memberikan dukungan penuh.

Sukabumi, Juli 2025

Rifqi Ramdhani

Nusa Putra
UNIVERSITY
2002
AMOR DEUS, PARENTIUM, CONCERVIS

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik UNIVERSITAS NUSA PUTRA, saya yang bertanda tangan di bawah ini:

Nama : Rifqi Ramdhani
NIM 20210040037
Program Studi : Teknik Informatika
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Nusa Putra **Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul: PENERAPAN METODE EVENT DRIVEN DESIGN MENGGUNAKAN *RABBITMQ* DAN *DOCKER* UNTUK OPTIMALISASI SKALABILITAS (STUDI KASUS: E-COMMERCE) beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Nusa Putra berhak menyimpan, mengalih media/format- kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: SUKABUMI
Pada tanggal: 14 Juli 2025
Yang menyatakan

RIFQI RAMDHANI

DAFTAR ISI

Nusa Putra UNIVERSITY 2009 AMOR DEUS, PARENTIUM, CONCERVIS

Nusa Putra UNIVERSITY 2009 AMOR DEUS, PARENTIUM, CONCERVIS

Nusa Putra
UNIVERSITY
2002
AMOR DEUS, PARENTIUM, CONCERVIS

DAFTAR TABEL

Nusa Putra UNIVERSITY 2002 AMOR DEUS, PARENTIUM, CONCERVIS

DAFTAR GAMBAR

Nusa Putra UNIVERSITY
AMOR DEUS, PARENTIUM, CONCERVIS

Nusa Putra
UNIVERSITY
2002
AMOR DEUS, PARENTIUM, CONCERVIS

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada era digital saat ini, e-commerce telah menjadi salah satu sektor yang berkembang paling pesat, dengan volume transaksi yang terus meningkat setiap tahunnya. Pertumbuhan ini didorong oleh kemudahan akses internet, meningkatnya penetrasi perangkat mobile, serta perubahan perilaku konsumen yang lebih memilih transaksi daring. Akibatnya, kebutuhan akan sistem yang mampu menangani permintaan dalam skala besar dan kompleks menjadi semakin krusial, terutama dalam pengolahan data secara real-time. Menurut penelitian (Nida et al., 2024), e-commerce telah menjadi bagian integral dari perekonomian digital, dengan tingkat pertumbuhan tahunan yang diperkirakan mencapai 53,23% di Indonesia[1].

Seiring dengan meningkatnya skala transaksi dan jumlah pengguna, tantangan dalam pengelolaan infrastruktur teknologi semakin nyata. Salah satu tantangan utama adalah bagaimana sistem dapat menangani lonjakan permintaan yang tinggi tanpa mengalami bottleneck atau penurunan performa. Sebelum pendekatan Event-Driven Design (EDD) diadopsi, sistem e-commerce umumnya mengandalkan arsitektur monolitik atau request-response berbasis REST API. Pada arsitektur ini, semua komponen aplikasi (UI, logika bisnis, database) terintegrasi dalam satu codebase, sehingga setiap permintaan harus diproses secara berurutan oleh server pusat. Pendekatan ini rentan mengalami bottleneck karena ketergantungan pada sumber daya terpusat, terutama saat terjadi lonjakan traffic.

Berdasarkan keterbatasan tersebut, metode event-driven design menawarkan pendekatan yang lebih efisien dalam mengelola interaksi dan komunikasi antar komponen sistem. Pendekatan ini mampu meningkatkan responsivitas dan skalabilitas pada aplikasi e-commerce. Seperti yang disebutkan oleh Munshi (2023), " Arsitektur analitik big data e-commerce yang tangguh memerlukan

kemampuan pemrosesan real-time untuk memastikan pengalaman pengguna yang lancar dalam skenario lalu lintas tinggi." [2]. Hal ini menunjukkan bahwa arsitektur yang tangguh sangat diperlukan untuk menangani volume transaksi yang besar, terutama pada saat puncak aktivitas pengguna.

Dalam implemetasi event-driven design, teknologi seperti *RabbitMQ* dan containerization (*Docker*) muncul sebagai solusi inovatif. *RabbitMQ* adalah message broker berbasis protokol AMQP yang memungkinkan komunikasi asinkron antar komponen sistem melalui mekanisme publish-subscribe. Dengan *RabbitMQ*, setiap perubahan sistem (misalnya: pesanan dibuat, pembayaran berhasil) dikirim sebagai event yang dapat diproses secara independen oleh layanan terkait. Sementara itu, containerization menggunakan *Docker* memungkinkan pengemasan layanan ke dalam lingkungan terisolasi (container) yang ringan, sehingga memudahkan replikasi dan skalabilitas infrastruktur.

Meskipun *RabbitMQ* telah banyak digunakan untuk mengoptimalkan komunikasi antar microservices dan aplikasi berbasis event-driven, implementasinya dalam skenario e-commerce berskala besar belum dieksplorasi secara menyeluruh Penelitian sebelumnya lebih banyak berfokus pada evaluasi performa *RabbitMQ* dalam lingkungan simulasi atau kasus umum, tanpa secara spesifik menyoroti perannya dalam menangani lonjakan request dan volume data yang tinggi di aplikasi e-commerce. Gasparyan et al. (2020) menyebutkan, " Mekanisme komunikasi berbasis peristiwa mengoptimalkan jaringan yang berpusat pada layanan dengan mengurangi beban protokol dan meningkatkan waktu pemrosesan selama kondisi beban tinggi." [3]. Kutipan ini menunjukkan potensi efisiensi yang dapat dicapai melalui mekanisme komunikasi berbasis event-driven.

Selain *RabbitMQ*, *Docker* sebagai platform containerization, memberikan solusi tambahan untuk meningkatkan efisiensi dan fleksibilitas infrastruktur aplikasi. Dengan menggunakan *Docker*, layanan *RabbitMQ* dan komponen e-commerce lainnya dapat dikemas dalam container yang terisolasi, memungkinkan deployment yang lebih cepat, skalabilitas yang lebih baik, dan pengelolaan layanan

yang lebih mudah. *Docker* juga mendukung pengelolaan sumber daya yang lebih efisien, sehingga sangat relevan untuk lingkungan dengan volume data besar dan permintaan yang tinggi.

Penelitian sebelumnya telah memberikan landasan kuat untuk penerapan Event-Driven Design (EDD) dengan *RabbitMQ* dan *Docker* dalam konteks e-commerce. Gasparyan et al. (2020) dalam studinya membuktikan bahwa mekanisme komunikasi berbasis event-driven secara signifikan mengurangi overhead protokol dan meningkatkan waktu pemrosesan dalam kondisi beban tinggi, terutama pada sistem yang membutuhkan respons real-time. Temuan ini diperkuat oleh Rahmatulloh et al. (2022) yang mengevaluasi implementasi Event-Driven Architecture dalam sistem berbasis microservices dan menunjukkan bahwa pendekatan ini dapat meningkatkan skalabilitas serta performa sistem secara signifikan. Selain itu, Zuki et al. (2024) menyoroti bagaimana containerization berbasis *Docker* dapat meningkatkan efisiensi sistem event-driven, dengan hasil penelitian menunjukkan bahwa containerized event-driven microservice architecture mampu meningkatkan fleksibilitas deployment dan efisiensi pemanfaatan sumber daya hingga 35%. Hasil penelitian ini menjadi dasar untuk mengatasi tantangan teknis yang diidentifikasi dalam studi sebelumnya, seperti manajemen sumber daya dan penanganan kegagalan pesan.

Penelitian ini bertujuan untuk mengeksplorasi penerapan Event-Driven Design (EDD) menggunakan *RabbitMQ* sebagai message broker dalam menangani request dengan jumlah data yang besar pada aplikasi e-commerce. Fokus utama adalah pengukuran performa sistem dalam skenario high traffic dengan parameter seperti latensi, throughput, tingkat keberhasilan pengiriman pesan, dan efisiensi pemanfaatan sumber daya. Behl (2023) mencatat, " Meskipun teknologi memfasilitasi pertumbuhan e-commerce, keberhasilannya bergantung pada adopsi arsitektur yang skalabel secara efektif, yang dapat mengatasi baik sisi positif maupun negatif dari operasi bervolume tinggi." [4]. Ini menegaskan pentingnya arsitektur yang skalabel untuk mendukung pertumbuhan e-commerce.

Berdasarkan latar belakang tersebut, penulis tertarik untuk melakukan penelitian dengan judul **"PENERAPAN METODE EVENT DRIVEN DESIGN (EDD) MENGGUNAKAN *RABBITMQ* DAN *DOCKER* UNTUK OPTIMALISASI SKALABILITAS E-COMMERCE"**. Penulis berharap penelitian ini dapat memberikan kontribusi yang signifikan, baik secara teoritis maupun praktis, dalam pengembangan aplikasi e-commerce yang lebih efisien dan dapat menangani permintaan dalam skala besar dengan lebih baik.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang diajukan penulis, yaitu:

1. Bagaimana penerapan metode *Event Driven Design* (EDD) dengan *RabbitMQ* sebagai message broker dan *Docker* dapat membantu menangani request dengan jumlah data yang besar pada aplikasi e-commerce?
2. Bagaimana dampak penerapan *Event Driven Design* (EDD) menggunakan *RabbitMQ* dan *Docker* terhadap efisiensi proses transaksi pada e-commerce, terutama dalam menangani lonjakan permintaan selama event promosi seperti flash sale.
3. Bagaimana performa *RabbitMQ* yang di jalankan dalam *Docker* dalam menangani request dengan volume data yang tinggi di aplikasi e-commerce berdasarkan parameter, latensi, throughput, tingkat keberhasilan pengiriman pesan?

1.3 Batasan Masalah

Agar penelitian ini mencapai pada sasaran yang diinginkan, maka penulis membatasi permasalahan dalam penelitian ini sebagai Berikut:

1. Penelitian ini hanya akan fokus pada penerapan metode *Event Driven Design* (EDD) dengan menggunakan *RabbitMQ* sebagai message broker dan *Docker* untuk isolasi layanan.
2. Pengujian akan dilakukan pada lingkungan yang telah disimulasikan dengan skenario high traffic, menggunakan data dummy yang menyerupai transaksi e-commerce sebenarnya.
3. Evaluasi performa hanya akan mencakup parameter latensi, throughput, tingkat keberhasilan pengiriman pesan.
4. Fokus implementasi dan pengujian performa dalam penelitian ini hanya terbatas pada proses penambahan produk ke dalam keranjang (Add to Cart), tanpa mencakup proses checkout, validasi stok, atau penghapusan item.
5. Penelitian ini menggunakan sample data dari platform Kaggle berupa log aktivitas pengguna e-commerce, khususnya aktivitas *Add to Cart*, sebagai acuan dalam menyusun skenario pengujian. Data ini digunakan untuk mensimulasikan pola trafik pengguna secara lebih realistis, bukan berasal dari data transaksi asli aplikasi.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini sebagai berikut:

1. Menganalisis penerapan metode *Event Driven Design* (EDD) menggunakan *RabbitMQ* sebagai message broker dan *Docker* pada aplikasi e-commerce.
2. Mengevaluasi kinerja *RabbitMQ* dalam lingkungan *Docker* untuk menangani request dengan jumlah data yang besar pada skenario high traffic, dengan mengukur parameter seperti latensi, throughput, tingkat keberhasilan pengiriman pesan, dan efisiensi pemanfaatan sumber daya.
3. Mengidentifikasi dampak penerapan metode EDD terhadap responsivitas dan skalabilitas aplikasi e-commerce.

1.5 Manfaat penelitian

Manfaat yang diharapkan oleh peneliti dalam penelitian ini adalah sebagai berikut:

1. Manfaat Teoritis
   a) Berkontribusi pada pengembangan pengetahuan di bidang arsitektur perangkat lunak, khususnya dalam penerapan metode *Event Driven Design* dengan *RabbitMQ* sebagai *message broker* dan *Docker* sebagai pendukung skalabilitas.
   b) Menyediakan referensi bagi penelitian selanjutnya terkait optimalisasi kinerja aplikasi berbasis event-driven pada sektor e-commerce.
2. Manfaat Praktis
   a) Memberikan panduan teknis bagi pengembang aplikasi e-commerce dalam mengimplementasikan metode EDD dengan *RabbitMQ* dan *Docker* untuk menangani permintaan dengan jumlah data besar.
   b) Membantu pengembang aplikasi meningkatkan responsivitas dan skalabilitas sistem mereka dengan memanfaatkan teknologi containerization untuk menghadapi tantangan high traffic secara lebih efisien
   c) Memberikan masukan bagi pengambil keputusan dalam memilih arsitektur perangkat lunak yang tepat untuk mendukung pertumbuhan e-commerce dengan memanfaatkan kombinasi *RabbitMQ* dan *Docker*.

1.6 Sitematika Penulisan

Untuk memudahkan bagi pembaca dalam menganalisa dan memahami hasil dari penelitian yang dilakukan penulis, maka penulis membuat suatu sistematika penulisan yang dibagi atas beberapa bab sebagai Berikut:

BAB I: PENDAHULUAN

Berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Membahas penelitian terkait dan kerangka pemikiran yang mendasari penelitian ini.

BAB III: METODOLOGI PENELITIAN

Membahas tahapan penelitian, metode pengumpulan data, dan skenario pengujian yang digunakan.

BAB IV: HASIL DAN PEMBAHASAN

Menyajikan hasil penelitian dan analisis terhadap ketercapaian tujuan penelitian.

BAB V: PENUTUP

Berisi ringkasan hasil penelitian, kesimpulan, dan rekomendasi untuk penelitian selanjutnya.

Nusa Putra UNIVERSITY 2002 AMOR DEUS, PARENTIUM, CONCERVIS

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan melalui pengujian performa sistem e-commerce pada proses *Add to Cart* menggunakan dua pendekatan arsitektur, yaitu *Remote Procedure Call* (RPC) dan *Event Driven Design* (EDD), maka dapat disimpulkan hal-hal berikut:

1. Penerapan arsitektur *Event Driven Design* (EDD) menggunakan *RabbitMQ* dan *Docker* memberikan dampak signifikan terhadap efisiensi sistem. Hal ini dibuktikan dengan latensi yang lebih rendah dan throughput yang lebih tinggi pada semua skenario pengujian, khususnya saat menangani lonjakan trafik seperti pada 50 virtual user.
2. Penggunaan *RabbitMQ* sebagai message broker dan *Docker* sebagai platform isolasi layanan terbukti membantu sistem dalam menangani permintaan berskala besar. Sistem EDD mampu memproses permintaan secara asinkron dan paralel, sehingga tetap stabil meskipun beban tinggi.
3. Berdasarkan parameter latensi, throughput, dan error rate, performa arsitektur EDD menunjukkan hasil yang lebih baik dibanding RPC. Tidak ditemukan error (error rate 0%) pada kedua sistem, namun RPC menunjukkan peningkatan waktu respons secara drastis saat beban tinggi, sedangkan EDD tetap stabil dan efisien.

Dengan demikian, arsitektur *Event Driven Design* dinilai lebih cocok untuk sistem e-commerce berskala besar yang membutuhkan ketahanan dan skalabilitas dalam menangani trafik tinggi secara real-time.

5.2 Saran

Berdasarkan hasil dan keterbatasan penelitian ini, penulis memberikan beberapa saran sebagai berikut:

1. Penelitian selanjutnya disarankan menguji proses checkout dan validasi stok, karena proses tersebut lebih kompleks dan kritis dalam transaksi e-commerce.
2. Sistem dapat diperluas ke microservices penuh, di mana masing-masing fitur (cart, product, order) dipisahkan secara fisik dan dihubungkan dengan event-driven communication, untuk melihat dampak skalabilitas yang lebih luas.
3. Penggunaan metode pengujian performa yang lebih beragam seperti stress test jangka panjang, penggunaan data real-world, atau simulasi trafik berbasis waktu nyata juga dapat memberikan hasil yang lebih representatif.
4. Untuk implementasi di dunia nyata, sistem event-driven perlu dilengkapi dengan mekanisme fallback, retry, dan monitoring, agar sistem lebih andal dalam produksi.

Nusa Putra
UNIVERSITY
2002
AMOR DEUS, PARENTIUM, CONCERVIS

DAFTAR PUSTAKA

[1] S. Nida, A. Nurhakim, J. M. Noor Isiqamah, and Nuraini, "ANALISIS PERKEMBANGAN TOKO ONLINE (E-COMMERCE) DI INDONESIA," *Jurnal Bisnis Digital (J-BisDig)*, vol. 2, no. 1, pp. 126–137, May 2024, doi: 10.52060/j-bisdig.v2i1.2180.

[2] A. Munshi, A. Alhindi, T. M. Qadah, and A. Alqurashi, "An Electronic Commerce Big Data Analytics Architecture and Platform," *Applied Sciences*, vol. 13, no. 19, p. 10962, Oct. 2023, doi: 10.3390/app131910962.

[3] M. Gasparyan, E. Schiller, A. Marandi, and T. Braun, "Communication Mechanisms for Service-Centric Networking," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, Jan. 2020, pp. 1–8. doi: 10.1109/CCNC46108.2020.9045531.

[4] A. Behl and J. Z. Zhang, "Guest editorial: Role of technology in E-commerce: bright and the dark side," *South Asian Journal of Business Studies*, vol. 12, no. 3, pp. 317–320, Aug. 2023, doi: 10.1108/SAJBS-08-2023-460.

[5] A. Rahmatulloh, F. Nugraha, R. Gunawan, and I. Darmawan, "Event-Driven Architecture to Improve Performance and Scalability in Microservices-Based Systems," in *2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, IEEE, Nov. 2022, pp. 01–06. doi: 10.1109/ICADEIS56544.2022.10037390.

[6] S. Z. M. Zuki, R. Mohamad, and N. A. Saadon, "Containerized Event-Driven Microservice Architecture," *Baghdad Science Journal*, vol. 21, no. 2(SI), p. 0584, Feb. 2024, doi: 10.21123/bsj.2024.9729.

[7] S. Khriji, Y. Benbelgacem, R. Chéour, D. El Houssaini, and O. Kanoun, "Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks," *J Supercomput*, vol. 78, no. 3, pp. 3374–3401, Feb. 2022, doi: 10.1007/s11227-021-03955-6.

[8] N. Islavath, "The Power of Docker: Containerization for Efficient Software Development and Deployment," *International Journal of Science and Research (IJSR)*, vol. 9, no. 11, pp. 1748–1751, Nov. 2020, doi: 10.21275/SR201226085354.

[9] D. P. Divakaran, "Transforming the Software Development Lifecycle with Docker Containers: A Comparative Study," *International Journal for Multidisciplinary Research (IJFMR)*, vol. 4, no. 1, pp. 1–7, Jan.–Feb. 2022, doi: 10.36948/ijfmr220137026.

[10] I. Cahya Gumilar, A. Sujjada, and K. Kamdan, “Implementasi Arsitektur Microservice pada Sistem Informasi HRD Berbasis WEB,” vol. 6, no. 5, 2025, doi: 10.38035/jemsi.v6i5.

[11] M. Rajasinghe, "Adoption challenges of CI/CD methodology in software development teams," *TechRxiv*, preprint, Sep. 28, 2021. doi: 10.36227/techrxiv.16681957.v1

[12] G. Fu, Y. Zhang, and G. Yu, “A Fair Comparison of Message Queuing Systems,” *IEEE Access*, vol. 9, pp. 421–432, 2021, doi: 10.1109/ACCESS.2020.3046503.

[13] M. Adlan Al Hawari Nasution and E. Suryana, “RANCANGAN MEDIA PEMBELAJARAN BERUPA APLIKASI AUGMENTED REALITY BERBASIS ANDROID,” 2023.

[14] Y. Aryani, I. Aqil, and B. Paramita, “Penerapan Unified Modeling Language (UML) pada Digitalisasi Sistem Informasi Perpustakaan,” *Digital Transformation Technology*, vol. 4, no. 2, pp. 1032–1040, Jan. 2025, doi: 10.47709/digitech.v4i2.5153.

[15] D. Rahmayanty, R. R. Guk Guk, B. D. I. Cahya, and M. Regilsa, “Peran Orang Tua Dalam Mengaplikasikan Internet Sebagai Media Pendidikan Bagi Anak,” *Jurnal Pendidikan dan Konseling*, vol. 5, no. 6, pp. 45–55, Dec. 2023. doi: 10.31004/jpdk.v5i6.20182

[16] T. P. Nguyen, “Add to cart dataset,” *Kaggle*, 2022. [Online]. Available: https://www.kaggle.com/datasets/teajay/global-superstore-add-to-cart