JURNAL ILMIAH SISTEM INFORMASI (JUSI)
VOL 5, No. 3, November 2026, pp. 195-204
p-ISSN : 2803-1507 (media cetak) | e-ISSN : 2803-1531 (media online)
Link Publikasi: https://ejurnal.provisi.ac.id/index.php/JUISI/issue/view/175

Performance Efficiency Analysis of a C++-Based POS and Inventory Management System Using Synthetic Transaction Workloads

Melda Dwi Susanti*[1], Aliyu Ibrahim Ahmad[2], Zaskiyah Sofarina[3], Najwa Amalia Khoir[4], Imam Prayogo Pujiono[5], Ahmad Mayere Abdullahi[6]
[1,2,3,4,5] Fakultas Science dan Teknologi, Universitas Islam Negeri K.H Abdurrahman Wahid Pekalongan
Jl. Kusuma Bangsa No. 9, 51141, Pekalongan, Jawa Tengah, Indonesia.
[6] School of Engineering & Technology, Noida International University, Delhi
Plot 1, Sector-17 A, Yamuna Expressway, Sector 17A, Gautam Budh Nagar, Uttar Pradesh - 203201, India.
E-mail: melda.dwi.susanti25017@mhs.uingusdur.ac.id

ARTICLE INFO

Article History:

Received: 15-05-2026
Revised: 16-06-2027
Accepted: 25-07-2027
Published: 28-10-2026

ABSTRACT

The increasing digitization of retail processes has further escalated the need to have effective software systems that are able to ensure real time transaction integrity, especially in the resource limited local e-commerce settings. Nonetheless, the solutions currently available are based on the managed programming languages, which add a computational overhead and result in unpredictable runtime behavior. This paper explores the performance efficiency of a cashier and inventory management system that is implemented with C++ and focuses on deterministic memory management and a low-level control over the system.This study used an experimental approach, which involved the use of synthetic transaction workloads to simulate retail transaction scenarios, under different conditions, between 1,000 and 100,000 transactions. The main performance indicators, such as execution time, throughput, latency, and memory stability, were systematically measured. The findings prove that the C++-based system has a high throughput, lower latency and constant memory use at all the workload levels, which connotes good scalability and efficiency.The results underscore the benefits of systems-level programming in reducing overhead and enhancing performance in limited environments. This research study adds to the expanding literature supporting low-level optimization of localized e-commerce systems and offers an empirical model of the creation of high-performance retail systems.

Keywords: C++, Point of Sale (POS) systems, inventory management, work efficiency, and local retail businesses.

Abstrak

Meningkatnya digitalisasi proses ritel semakin memperkuat kebutuhan akan sistem perangkat lunak yang efektif dan mampu menjamin integritas transaksi secara *real-time*, terutama dalam lingkungan *e-commerce* lokal yang memiliki keterbatasan sumber daya. Meskipun demikian, solusi yang tersedia saat ini didasarkan pada bahasa pemrograman yang dikelola, yang menambah beban komputasi dan mengakibatkan perilaku runtime yang tidak dapat diprediksi. Makalah ini mengeksplorasi efisiensi kinerja sistem kasir dan manajemen inventaris yang diimplementasikan dengan C++ dan berfokus pada manajemen memori deterministik serta kontrol tingkat rendah atas sistem. Studi ini menggunakan pendekatan eksperimental, yang melibatkan penggunaan beban kerja transaksi sintetis untuk mensimulasikan skenario transaksi ritel, di bawah kondisi yang berbeda, antara 1.000 hingga 100.000 transaksi. Indikator kinerja utama, seperti waktu eksekusi, *throughput*, latensi, dan stabilitas memori, diukur secara sistematis. Temuan menunjukkan bahwa sistem berbasis C++ memiliki *throughput* tinggi, latensi lebih rendah, dan penggunaan memori yang konstan

DOI: https://doi.org/10.51903/rahm1q97

p-ISSN : 2803-1507 e-ISSN : 2803-1531

pada semua tingkat beban kerja, yang menandakan skalabilitas dan efisiensi yang baik. Hasil ini menggarisbawahi manfaat pemrograman tingkat sistem dalam mengurangi *overhead* dan meningkatkan kinerja di lingkungan yang terbatas. Penelitian ini menambah literatur yang terus berkembang yang mendukung optimalisasi tingkat rendah dari sistem *e-commerce* yang dilokalkan dan menawarkan model empiris pembuatan sistem ritel berkinerja tinggi.

Kata Kunci: *C++, sistem kasir (POS), manajemen inventaris, efisiensi kerja, ritel lokal*

1. PENDAHULUAN

Perkembangan teknologi *digital* telah mendorong perubahan besar dalam sistem perdagangan modern, khususnya pada sektor *e-commerce* dan layanan transaksi berbasis komputer. Peningkatan aktivitas perdagangan digital menyebabkan pelaku usaha semakin bergantung pada sistem transaksi yang mampu bekerja secara cepat, stabil, dan efisien dalam mengelola data penjualan serta inventaris produk. Dalam lingkungan bisnis yang kompetitif, sistem *point-of-sale* (POS) dan manajemen inventaris tidak lagi hanya berfungsi sebagai alat administrasi transaksi, tetapi juga menjadi bagian penting dalam menjaga efektivitas operasional dan kualitas layanan pelanggan [1]. Implementasi sistem transaksi *digital* yang terintegrasi memungkinkan sinkronisasi stok dilakukan secara *real-time* sehingga aktivitas penjualan dan pembaruan inventaris dapat berjalan lebih akurat. Selain meningkatkan efisiensi operasional, sistem POS *modern* juga berperan dalam mempercepat proses transaksi dan meminimalkan kesalahan pencatatan *data* barang pada lingkungan *retail digital*. Penelitian oleh Purwasih et al. [2] menjelaskan bahwa pengelolaan inventaris berbasis sistem informasi mampu meningkatkan efektivitas pengelolaan data serta mempercepat proses administrasi transaksi pada perusahaan ritel.

Terlepas dari evolusi berkelanjutan dari sistem transaksi digital, banyak implementasi *e-commerce* lokal masih dihadapkan pada banyak keterbatasan kinerja karena *volume* transaksi meningkat pesat. Masalah seperti penundaan dalam pemrosesan transaksi, peningkatan latensi, pemanfaatan memori yang tidak konsisten, dan perbedaan dalam sinkronisasi inventaris sering diamati dalam sistem ritel *digital*. Kondisi buruk seperti itu dapat secara signifikan mempengaruhi kualitas layanan dan respons sistem ketika digunakan dalam pengaturan operasional yang ditandai dengan *volume* transaksi yang tinggi [3], [4]. Stabilitas sistem transaksi *real-time* merupakan elemen penting dalam memastikan kontinuitas operasional dalam lanskap ritel *modern* [5]. Selanjutnya, dijelaskan bahwa optimalisasi manajemen data transaksi di samping efisiensi pemrosesan sistem mewakili dimensi penting dalam memperkuat stabilitas operasional perusahaan ritel yang berpusat pada data. Tantangan ini menggarisbawahi meningkatnya kebutuhan akan sistem transaksi dengan efisiensi komputasi tinggi untuk memfasilitasi kemajuan *e-commerce* lokal yang berkelanjutan.[6]

Sejumlah penelitian sebelumnya telah mengeksplorasi peningkatan inventaris *digital* dan sistem transaksi dengan maksud untuk meningkatkan efektivitas operasional perusahaan [7]. Merancang sistem inventaris berbasis web yang bertujuan untuk meningkatkan efisiensi penjualan di perusahaan kecil dan menengah. Selain itu, penelitian berjudul "Desain dan Pengembangan *Sistem Point of Sale* Berbasis Web untuk Efisiensi Bisnis Ritel" (2025) menekankan pentingnya integrasi sistem transaksi dan inventaris dalam mendorong efisiensi kerangka kerja ritel *digital.* Namun demikian, sebagian besar upaya penelitian sebelumnya berkonsentrasi pada pengembangan sistem berbasis web dan belum menilai kinerja sistem secara ketat melalui metodologi *benchmarking* komputasi. Evaluasi parameter seperti waktu eksekusi, *throughput*, latensi, dan pemanfaatan memori tetap relatif tidak mencukupi, terutama mengenai penerapan sistem *checkout* dan inventaris berbasis C++ dalam konteks *e-commerce* lokal. Penelitian yang berpusat pada *benchmarking* sistem komputasi menunjukkan bahwa pengujian beban kerja dapat menawarkan representasi yang lebih otentik dari perilaku sistem di berbagai tingkat beban kerja. Penelitian sebelumnya menjelaskan bahwa pengukuran berdasarkan *throughput* dan latensi merupakan metodologi yang signifikan untuk menilai kinerja sistem pemrosesan data modern [8]. Selain itu, menguatkan bahwa efisiensi memori dan stabilitas *runtime* adalah penentu penting dalam kemajuan sistem komputasi berkinerja tinggi [9].

Bahasa pemrograman C++ merupakan alternatif yang signifikan dalam desain sistem transaksi *digital*, karena kemampuannya untuk manajemen memori tingkat rendah, efisiensi *runtime* yang unggul, dan stabilitas kinerja yang ditingkatkan dibandingkan dengan banyak bahasa pemrograman terkelola lainnya. Adopsi paradigma pemrograman tingkat rendah memungkinkan sistem untuk meminimalkan *overhead runtime* dan mengoptimalkan efisiensi pemrosesan transaksi. Menjelaskan bahwa sistem ritel *digital* yang didasarkan pada kinerja tinggi memerlukan strategi pengembangan yang dapat mempertahankan konsistensi *throughput* dan efisiensi pemanfaatan memori di seluruh spektrum keadaan operasional [10]. Selanjutnya, menunjukkan bahwa stabilitas akses data dan efisiensi transaksi merupakan penentu penting dalam memfasilitasi evolusi

p-ISSN : 2803-1507 e-ISSN : 2803-1531

sistem *e-commerce modern* yang dapat diskalakan dan responsif [11]. Mengingat tantangan ini, penelitian ini berusaha untuk menilai efisiensi kinerja mesin kasir dan sistem inventaris berdasarkan C++ dalam konteks *e-commerce* lokal melalui metodologi eksperimental komputasi. Evaluasi dilakukan dengan menggunakan simulasi beban kerja sintetis dengan variasi *volume* transaksi untuk memastikan waktu eksekusi, *throughput*, latensi, dan stabilitas pemanfaatan memori sistem. penelitian ini bercita-cita untuk menghasilkan kontribusi praktis dan teoritis untuk kemajuan sistem transaksi *digital* yang lebih efisien, stabil, dan dapat diskalakan dalam kerangka kerja *e-commerce* lokal yang berorientasi pada kinerja tinggi.

2. TINJAUAN PUSTAKA

2.1. Penelitian Sebelumnya tentang Sistem POS dan Inventaris

Investigasi ilmiah ekstensif ke dalam sistem *point-of-sale* (POS) dan manajemen inventaris telah dilakukan untuk meningkatkan efisiensi operasional dalam lanskap ritel *digital*. Menjelaskan bahwa adopsi sistem POS terintegrasi dapat secara signifikan meningkatkan efisiensi transaksi dan presisi inventaris di perusahaan ritel *modern* [1]. Lebih lanjut, penelitian yang dilakukan menunjukkan bahwa penyebaran sistem inventaris *digital* memfasilitasi pengelolaan data stok yang dipercepat dan sinkronisasi transaksi secara *real-time* [2]. Selain itu, merancang sistem inventaris berbasis web yang bertujuan untuk meningkatkan efisiensi penjualan dalam usaha kecil dan menengah [7]. Namun demikian, sebagian besar penelitian sebelumnya berpusat pada peningkatan fungsionalitas sistem dan telah mengabaikan penilaian kinerja komprehensif menggunakan metodologi *benchmarking* komputasi.

2.2. Perbandingan Sistem dan Penelitian Skalabilitas

Benchmarking kinerja merupakan metodologi penting untuk menilai stabilitas dan efisiensi operasional sistem transaksi *digital* di berbagai tingkat beban kerja. Penelitian sebelumnya menjelaskan bahwa perbandingan berbasis beban kerja mampu memberikan representasi otentik dari perilaku sistem dalam keadaan operasional yang berbeda [8]. Selain itu, menegaskan bahwa pengujian skalabilitas sangat penting untuk memastikan bahwa sistem mempertahankan *throughput* dan stabilitas kinerja di tengah beban transaksi yang meningkat [12]. Sebaliknya, menggarisbawahi pentingnya efisiensi memori dan stabilitas *runtime* dalam sistem komputasi berkinerja tinggi [9]. Penelitian yang dilakukan lebih lanjut menunjukkan bahwa beban kerja transaksi sintetis dapat secara efektif mensimulasikan aktivitas transaksi *digital* tanpa ketergantungan pada data operasional aktual [15].

2.3. Kesenjangan Penelitian

Analisis literatur yang ada mengungkapkan bahwa sebagian besar penelitian terutama berkonsentrasi pada pengembangan sistem transaksi berbasis web, gagal menilai kinerja mesin kasir berbasis C++ dan sistem inventaris melalui lensa benchmarking komputasi. Evaluasi parameter penting seperti waktu eksekusi, *throughput*, latensi, dan pemanfaatan memori dalam konteks *e-commerce* lokal masih relatif belum dijelajahi. Akibatnya, penelitian ini bertujuan untuk fokus pada analisis efisiensi kinerja mesin kasir dan sistem inventaris berbasis C++ menggunakan simulasi beban kerja transaksi sintetis untuk mencapai penilaian kinerja yang lebih skalabel dan otentik.

3. METODE PENELITIAN

3.1. Desain Penelitian

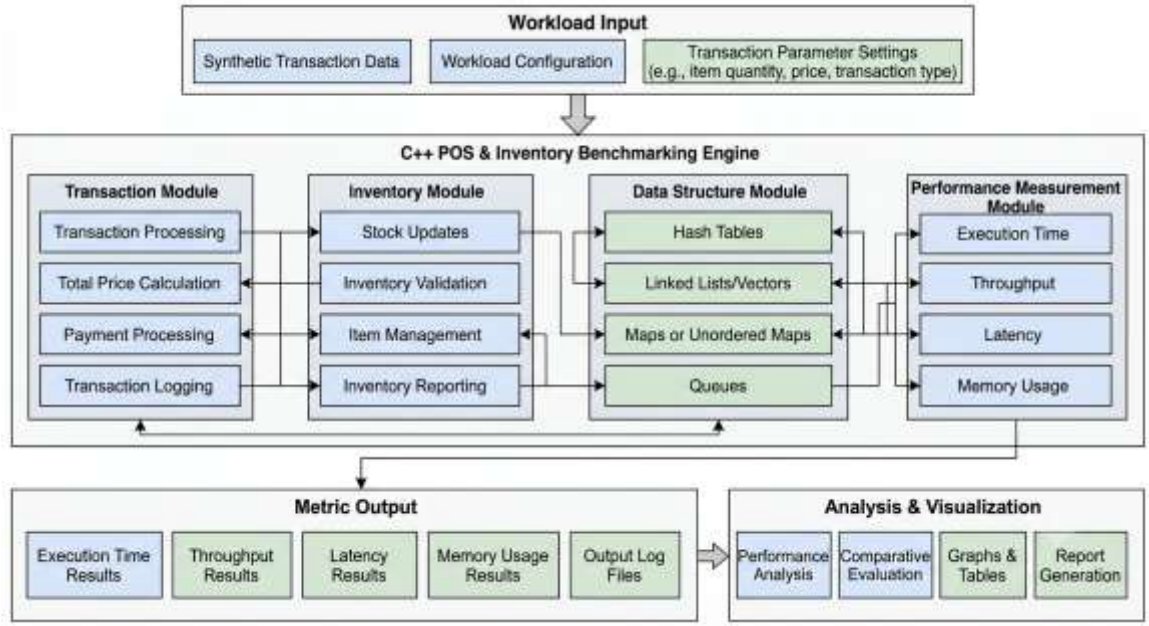
Penelitian ini menggunakan metodologi kuantitatif yang menggunakan teknik eksperimental komputasi untuk meneliti efisiensi kinerja sistem *checkout* dan inventaris berbasis C++ dalam konteks *e-commerce* lokal. Pemilihan metodologi eksperimental didasarkan pada kapasitasnya untuk memfasilitasi penilaian kinerja sistem yang objektif dan terukur melalui simulasi transaksi di berbagai intensitas beban kerja. Di bidang penelitian sistem komputasi modern, metodologi perbandingan eksperimental diadopsi secara ekstensif untuk menilai konsistensi kinerja, pemanfaatan sumber daya, dan skalabilitas sistem dalam berbagai keadaan operasional [13]. Penekanan utama penelitian berpusat pada kemahiran sistem dalam mempertahankan kinerja yang stabil di tengah peningkatan yang signifikan dalam *volume* transaksi. Akibatnya, penelitian ini menerapkan metodologi evaluasi kinerja berbasis beban kerja untuk menyelidiki korelasi antara beban transaksi yang meningkat dan fluktuasi kinerja sistem. Metodologi paralel juga digunakan dalam penelitian perbandingan aplikasi *e-commerce* untuk menilai efisiensi algoritmik dan stabilitas sistem pada berbagai besaran kumpulan data [14].

3.2. Implementasi Sistem dan Pengaturan Eksperimental

Sistem yang dirancang dalam pertanyaan ini dirancang untuk mereplikasi aktivitas transaksi kasir dan sinkronisasi inventaris secara *real time.* Implementasi sistem dijalankan menggunakan bahasa pemrograman

p-ISSN : 2803-1507 e-ISSN : 2803-1531

C++, yang dikenal karena kemampuan kontrol memori tingkat rendah, manajemen memori deterministik, dan efisiensi *runtime* yang unggul dibandingkan dengan beberapa bahasa pemrograman terkelola lainnya. Pemanfaatan C++ dianggap kondusif untuk meminimalkan *overhead* komputasi dan menjaga stabilitas kinerja sementara sistem secara bersamaan memproses *volume* transaksi yang besar [13]. Kerangka arsitektur dari sistem *benchmarking* berbasis C++ yang digunakan dalam penyelidikan ini diilustrasikan pada Gambar 1. Skema menggambarkan aliran operasional beban kerja transaksi sintetis, pengelolaan komponen transaksi dan inventaris, implementasi struktur data penting, dan metodologi untuk menilai metrik kinerja sistem penting, termasuk waktu eksekusi, *throughput*, latensi, dan pemanfaatan memori.



Gambar 1. Arsitektur Sistem Benchmarking POS dan Inventory Berbasis C++**.**

Eksperimen dijalankan menggunakan beban kerja transaksi sintetis yang direkayasa untuk mewakili aktivitas transaksi dalam lingkungan *e-commerce* lokal. *Data* transaksi sintetis memfasilitasi simulasi proses pembelian, pembaruan stok inventaris, dan sinkronisasi data otomatis. Adopsi metodologi beban kerja sintetis dimotivasi oleh penyediaan fleksibilitasnya dalam mengatur besarnya kumpulan data, pola transaksi, dan intensitas beban kerja tanpa ketergantungan pada data operasional nyata yang ditandai dengan aksesibilitas terbatas atau masalah privasi [15]. Setiap transaksi sintetis yang digunakan dalam penelitian ini dibuat dengan cermat untuk merangkum operasi dasar yang dilakukan dalam sistem *Point of Sale* (POS), mencakup proses yang terkait dengan pemilihan produk, pembaruan stok inventaris, perhitungan nilai transaksi total, dan pencatatan data transaksi.

Prosedur evaluasi dijalankan melalui eksekusi metodis transaksi berurutan, di mana setiap transaksi diproses secara linier untuk menegakkan integritas konsistensi pengukuran kinerja dalam sistem. Selanjutnya, mekanisme pembaruan inventaris otomatis diterapkan setelah keberhasilan pemrosesan setiap transaksi untuk secara akurat mensimulasikan sinkronisasi tingkat stok dalam kerangka transaksi digital [15], [20]. Adopsi metodologi beban kerja berurutan dibenarkan oleh kapasitasnya untuk menghasilkan hasil perbandingan yang lebih stabil dan dapat diukur dalam penilaian kemanjuran sistem transaksi digital [12].

Penelitian ini berfokus pada simulasi benchmarking performa sistem menggunakan synthetic transaction workload dan tidak melibatkan data operasional ritel secara langsung. Dalam analisis ini, beban kerja transaksi dikategorikan menjadi tiga tingkatan utama, yaitu beban kerja rendah 1.000 transaksi, beban kerja menengah 10.000 transaksi, dan beban kerja tinggi 100.000 transaksi. Setiap skenario eksperimental dieksekusi selama tiga percobaan untuk meningkatkan keandalan hasil eksperimen dan mengurangi potensi bias pengukuran. Pendekatan eksperimental berulang ini secara konvensional digunakan dalam penelitian

p-ISSN : 2803-1507 e-ISSN : 2803-1531

membandingkan kinerja sistem untuk memastikan bahwa hasil pengukuran stabil dan mencerminkan kondisi operasional otentik [12]. Penelitian ini dilakukan dengan menggunakan perangkat dengan spesifikasi berikut:

- Sistem Operasi: Windows 11 Pro
- Processor: Intel Core i5
- RAM: 8 GB
- Penyimpanan: SSD 512 GB
- Compiler: GCC 13.2

Keseluruhan implementasi sistem dan prosedur pengujian dilakukan dengan menggunakan *Visual Studio Code* sebagai *Integrated Development Environment* (IDE). Sepanjang proses *benchmarking*, hanya aplikasi yang berhubungan langsung dengan eksperimen yang dijalankan untuk menegakkan konsistensi dalam penggunaan CPU dan memori perangkat.

3.3. Performance Evaluation Metrics

Metrik evaluasi kinerja yang digunakan dalam penelitian ini mencakup waktu eksekusi, *throughput,* latensi, dan pemanfaatan memori sistem. Waktu eksekusi berfungsi sebagai metrik untuk mengukur durasi pemrosesan agregat transaksi; *throughput* digunakan untuk mengevaluasi volume transaksi yang dapat diproses per detik; sedangkan latensi digunakan untuk mengukur respons sistem selama fase eksekusi transaksi. Selain itu, pemanfaatan memori diteliti untuk menilai ketahanan alokasi sumber daya sistem di berbagai intensitas beban kerja. Parameter ini dianggap berperan dalam menilai efisiensi, skalabilitas, dan stabilitas sistem transaksi *digital* berkinerja tinggi [16], [17]. Selanjutnya, evaluasi pemanfaatan memori dilakukan melalui pengamatan konsumsi memori sistem selama proses benchmarking di setiap skenario beban kerja transaksi. Nilai penggunaan memori yang dilaporkan menandakan konsumsi memori rata-rata selama eksekusi setelah penyelesaian sistem dari fase inisialisasi awal. Metodologi ini digunakan untuk mengurangi dampak memori dasar sebelum pemrosesan transaksi. Selain itu, pemantauan penggunaan memori dijalankan secara berkala selama beban kerja untuk memastikan stabilitas alokasi memori dan konsistensi kinerja sistem selama proses transaksi [20], [23].

Prosedur benchmarking dijalankan menggunakan perpustakaan chrono dalam bahasa pemrograman C++, menggunakan metodologi *high resolution clock* untuk memfasilitasi pencatatan waktu eksekusi yang tepat. Pemantauan pemanfaatan memori dilakukan sepanjang simulasi transaksi untuk membedakan variasi konsumsi memori di setiap kategori beban kerja. Selanjutnya, semua hasil eksperimen menjadi sasaran analisis komparatif untuk memastikan korelasi antara peningkatan *volume* transaksi dan perubahan kinerja sistem. Metodologi *benchmarking* yang didasarkan pada penilaian beban kerja dianggap efektif untuk menilai skalabilitas sistem dan stabilitas kinerja dalam aplikasi transaksi *digital* dibawah skenario operasional yang beragam [20]. Secara konseptual, penelitian ini dimulai dengan perumusan beban kerja transaksi sintetis yang dirancang untuk mereplikasi aktivitas transaksi kasir dan modifikasi inventaris *real-time.* Selanjutnya, sistem menjalani pengujian menggunakan metrik kinerja yang mapan untuk menilai stabilitas dan efisiensi pemrosesan transaksi di berbagai intensitas beban kerja. Hasil pengujian selanjutnya disajikan dalam format tabel dan representasi grafis untuk meningkatkan analisis korelasi antara beban transaksi yang diperbesar dan kinerja sistem. Metodologi *benchmarking* berdasarkan beban kerja dianggap menghasilkan penilaian yang lebih otentik dari sistem transaksi *digital modern* [7], [8].

4. HASIL DAN DISKUSI

Penilaian kinerja sistem dilakukan untuk memastikan efisiensi sistem *checkout* dan inventaris berbasis C++ dalam mengelola berbagai tingkat beban kerja transaksi dalam kerangka kerja *e-commerce* lokal. Protokol *benchmarking* dijalankan menggunakan tiga kategori utama beban kerja, khususnya 1.000 transaksi, 10.000 transaksi, dan 100.000 transaksi. Setiap penilaian diberikan selama tiga iterasi, dan hasil rata-rata digunakan sebagai dasar analisis untuk memastikan keandalan pengukuran kinerja sistem. Evaluasi dilakukan sesuai dengan metrik waktu eksekusi, *throughput*, latensi, dan konsumsi memori untuk mendapatkan pemahaman yang lebih holistik tentang stabilitas dan efisiensi sistem.

4.1. Hasil Pengujian *Execution Time dan Throughput*

Temuan empiris menunjukkan bahwa peningkatan dalam jumlah transaksi memberikan efek langsung pada waktu eksekusi sistem. Terlepas dari pengamatan ini, sistem terus menunjukkan konsistensi kinerja di semua beban kerja yang dikategorikan. Di bawah beban kerja minimal 1.000 transaksi, sistem mendokumentasikan durasi eksekusi rata-rata 12 ms, disertai dengan *throughput* 83.333 transaksi per detik. Karena beban kerja diperkuat menjadi 10.000 transaksi, durasi eksekusi meningkat menjadi 94 ms, dengan *throughput* 106.382

p-ISSN : 2803-1507 e-ISSN : 2803-1531

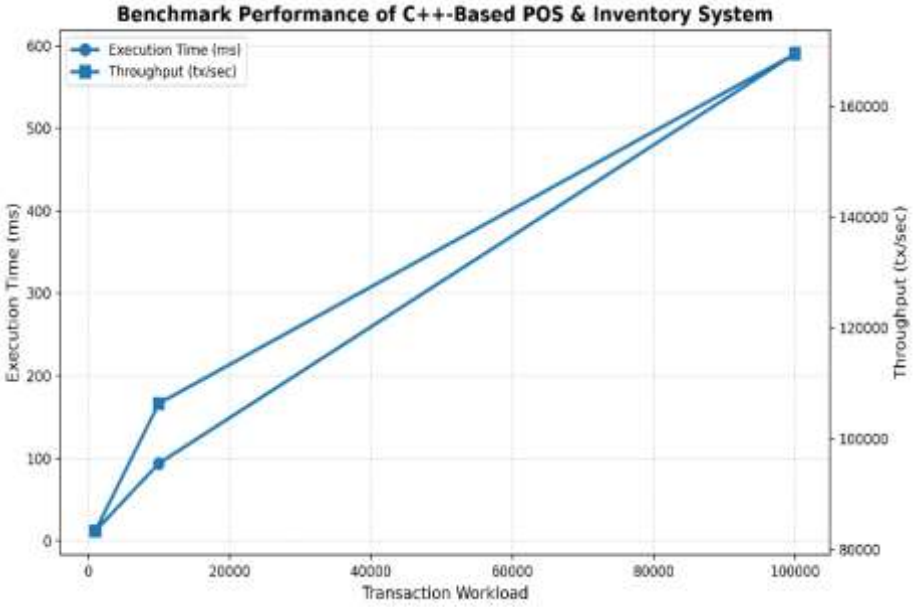
transaksi per detik. Sebaliknya, pada beban kerja 100.000 transaksi yang meningkat, durasi eksekusi dicatat pada 590 ms, dengan *throughput* mencapai 169.491 transaksi per detik.

Tabel 1. Hasil *Benchmarking* Kinerja Sistem POS dan Inventaris Berbasis C++

No	*Transaction Workload*	*Average Execution Time (ms)*	*Average Throughput (tx/sec)*	*Performance Status*
1	1,000 Transactions	12 ms	83,333 tx/sec	*Stable*
2	10,000 Transactions	94 ms	106,382 tx/sec	*Very Stable*
3	100,000 Transactions	590 ms	169,491 tx/sec	*Highly Scalable*

Menurut temuan empiris yang disajikan pada Tabel 2, latensi sistem tetap pada tingkat yang relatif minimal meskipun ada eskalasi yang nyata dalam *volume* transaksi. Di bawah kondisi operasional puncak, latensi rata-rata yang diamati hanya 4,7 ms, menunjukkan bahwa sistem mempertahankan kapasitasnya untuk memberikan respons transaksi yang cepat dan konsisten. Selain itu, pemanfaatan memori menunjukkan peningkatan bertahap seiring dengan meningkatnya beban kerja; Namun, itu tidak menunjukkan lonjakan konsumsi memori yang berlebihan. Temuan ini menyiratkan bahwa metodologi manajemen memori tingkat rendah yang digunakan dalam C ++ mahir dalam mempertahankan efisiensi alokasi memori, asalkan sistem berfungsi secara optimal [23] - [24]. Hasil tersebut menunjukkan bahwa sistem mampu mempertahankan peningkatan *throughput* meskipun jumlah transaksi meningkat secara signifikan. Kondisi ini menunjukkan bahwa pendekatan implementasi berbasis C++ memberikan efisiensi pemrosesan yang relatif stabil pada berbagai tingkat *workload*. Temuan ini sejalan dengan penelitian yang menunjukkan bahwa penggunaan C++ mampu memberikan efisiensi waktu komputasi yang lebih baik pada proses pengolahan data berskala besar [13].

Peningkatan *throughput* seiring bertambahnya jumlah transaksi dapat dipengaruhi oleh beberapa faktor teknis, seperti berkurangnya efek *initialization overhead* pada *workload* yang lebih besar, optimalisasi mekanisme perulangan (*loop execution*), serta peningkatan efisiensi *cache utilization* selama proses *benchmarking* berlangsung. Pada workload rendah, sebagian waktu eksekusi masih dipengaruhi oleh proses inisialisasi sistem dan alokasi memori awal, sedangkan pada *workload* yang lebih tinggi ...sistem mampu memproses transaksi secara lebih konsisten dan efisien dalam siklus eksekusi yang berulang [12], [20], [22]. Selain itu, stabilitas *throughput* yang dihasilkan menunjukkan bahwa sistem memiliki kemampuan skalabilitas yang cukup baik dalam menangani peningkatan aktivitas transaksi. Pada lingkungan *e-commerce* lokal, kemampuan mempertahankan *throughput* menjadi sangat penting karena berkaitan langsung dengan responsivitas sistem kasir dan sinkronisasi inventaris secara *real-time.* Penelitian mengenai *event-driven retail systems* juga menunjukkan bahwa peningkatan *throughput* dan pengurangan latensi memiliki pengaruh signifikan terhadap kualitas layanan transaksi digital pada lingkungan *retail modern* [18].



Gambar 2. Perbandingan Execution Time dan Throughput Berdasarkan Workload

Gambar 2 menggambarkan korelasi antara beban kerja transaksi yang diperbesar dan variasi waktu eksekusi dan *throughput* sistem. Temuan empiris menunjukkan bahwa *throughput* sistem terus meningkat bersamaan

p-ISSN : 2803-1507 e-ISSN : 2803-1531

dengan peningkatan *volume* transaksi, terlepas dari peningkatan waktu eksekusi secara bersamaan di bawah beban kerja yang meningkat. Pengamatan ini menunjukkan bahwa mesin kasir berbasis C++ dan sistem manajemen inventaris mampu mempertahankan stabilitas kinerja dan efisiensi pemrosesan transaksi di berbagai tingkat beban kerja. Kesimpulan ini sejalan dengan penelitian yang menjelaskan bahwa peningkatan *throughput* dan stabilitas latensi berfungsi sebagai metrik penting dalam menilai kinerja sistem transaksi *digital* yang didasarkan pada dinamika beban kerja [8].

4.2. Hasil Pengujian *Latency* dan Penggunaan Memori

Selain waktu eksekusi dan *throughput*, studi ini juga menilai latensi serta pemakaian memori sistem selama transaksi dilakukan. Hasil dari pengujian mengindikasikan bahwa peningkatan beban kerja berpengaruh terhadap kenaikan latensi dan penggunaan memori, namun peningkatan tersebut tetap berada pada level yang stabil dan terjaga. Hasil pengujian latency dan penggunaan memori sistem pada setiap kategori workload ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pengujian *Latency* dan Penggunaan Memori

Workload Transaksi	Latency (ms)	Penggunaan Memori (MB)
1.000	0,8	18
10.000	1,9	31
100.000	4,7	48

Menurut temuan empiris yang diperoleh dari pengujian yang dilakukan, latensi sistem tetap pada ambang yang relatif rendah, meskipun ada peningkatan signifikan dalam *volume* transaksi. Pada beban operasional puncak, latensi rata-rata yang dicatat hanya 4,7 milidetik, sehingga menandakan bahwa sistem terus memiliki kemampuan untuk memberikan respons transaksi yang cepat dan konsisten. Secara bersamaan, pemanfaatan memori telah mengalami peningkatan bertahap dalam korelasi dengan meningkatnya beban kerja; namun, hal tersebut tidak menunjukkan lonjakan signifikan dalam konsumsi memori yang berlebihan.

Pengamatan ini menunjukkan bahwa paradigma manajemen memori tingkat rendah yang digunakan dalam C++ secara efektif mempertahankan efisiensi alokasi memori selama sistem tetap beroperasi [23], [24]. Pernyataan ini menguatkan temuan yang diartikulasikan, yang menggambarkan pentingnya stabilitas transaksi dan sinkronisasi inventaris sebagai penentu penting dalam pelestarian kualitas layanan dalam kerangka ritel *digital*. Selanjutnya, penelitian yang berfokus pada perbandingan skalabilitas telah menjelaskan bahwa stabilitas penggunaan memori dan latensi merupakan metrik penting dalam penilaian kapasitas sistem untuk menjunjung tinggi kinerja di berbagai intensitas beban kerja [12], [19].

4.3. Analisis Efisiensi Sistem

Singkatnya, temuan menunjukkan bahwa mesin kasir dan sistem inventaris yang dikembangkan dalam C++ menunjukkan kapasitas untuk mempertahankan kinerja yang relatif konsisten di bawah beban transaksi yang meningkat. Efisiensi waktu eksekusi, ditambah dengan *throughput* yang tinggi dan latensi minimal, menunjukkan bahwa implementasi berbasis C++ memiliki potensi yang signifikan untuk aplikasi dalam pengaturan *e-commerce* lokal yang ditandai dengan permintaan transaksi yang terus meningkat [25]. Penerapan metodologi *benchmarking* komputasi dalam penelitian ini juga memberikan representasi yang lebih akurat dari perilaku sistem ketika dihadapkan dengan beban transaksi yang signifikan. Melalui simulasi beban kerja transaksi sintetis, penelitian ini mampu menilai korelasi antara peningkatan jumlah transaksi dan perubahan yang dihasilkan dalam kinerja sistem dengan cara yang lebih dapat diukur. Metodologi ini sejalan dengan temuan yang menggarisbawahi perlunya menilai kinerja sistem dalam kaitannya dengan variasi besarnya kumpulan data dan kerumitan beban kerja [14].

Tabel 3. Perbandingan Karakteristik Kinerja Implementasi Sistem POS

Programming Language	***Runtime Performance***	***Memory Efficiency***	***Scalability***	***Runtime Overhead***	***Suitability for High-Volume Transactions***
C++	*High*	*High*	*High*	*Low*	*Very Suitable*
Java	*Moderate–High*	*Moderate*	*High*	*Moderate*	*Suitable*
Python	*Moderate*	*Moderate–Low*	*Moderate*	*High*	*Limited*
PHP-Based Web System	*Moderate*	*Moderate*	*Moderate*	*Moderate–High*	*Suitable for Web Integration*
JavaScript (Node.js)	*Moderate–High*	*Moderate*	*High*	*Moderate*	*Suitable for Real-Time APIs*

Dari perspektif aplikasi praktis, temuan menunjukkan bahwa pemanfaatan C++ dapat berfungsi sebagai alternatif yang layak untuk kemajuan sistem transaksi digital yang memerlukan efisiensi memori, kinerja yang konsisten, dan potensi skalabilitas yang signifikan. Dengan karakteristik *throughput* yang dapat

p-ISSN : 2803-1507 e-ISSN : 2803-1531

diandalkan dan latensi minimal, sistem menunjukkan kapasitas untuk memfasilitasi aktivitas transaksi dan menyinkronkan inventaris secara *real-time* dalam pengaturan *e-commerce* lokal yang ditandai dengan volume transaksi yang tinggi [26].

Menurut analisis konseptual yang disajikan pada Tabel 3, implementasi berdasarkan C ++ menunjukkan keuntungan yang signifikan dalam hal efisiensi runtime dan manajemen memori jika dibandingkan dengan berbagai bahasa pemrograman terkelola lainnya. Atribut manajemen memori tingkat rendah yang melekat pada C ++ memfasilitasi pemeliharaan latensi yang berkurang dan stabilitas throughput yang ditingkatkan di lingkungan yang ditandai dengan volume transaksi yang tinggi. Sebaliknya, bahasa pemrograman seperti Python dan PHP umumnya mengeluarkan overhead runtime yang lebih tinggi karena ketergantungannya pada penerjemah dan lingkungan runtime terkelola. Namun demikian, bahasa seperti Java dan JavaScript memiliki keunggulan yang berbeda di bidang portabilitas, integrasi web, dan pengembangan sistem skala perusahaan [13] - [14], [18].

Meskipun hasil perbandingan menunjukkan kinerja sistem yang relatif konsisten di berbagai tingkat beban kerja transaksional, penyelidikan ini bukannya tanpa kendala. Penilaian yang dilakukan tidak mencakup simulasi multi-pengguna bersamaan, seluk-beluk kerangka kerja pemrosesan transaksi database, pertimbangan latensi jaringan, mekanisme untuk kontrol konkurensi, strategi penguncian, dan skenario kegagalan yang biasanya ditemui dalam konteks sistem transaksional yang ekspansif. Selanjutnya, pendekatan benchmarking terutama berfokus pada beban kerja transaksional sintetis dalam lingkungan yang terkendali, sehingga gagal merangkum seluk-beluk operasional yang ditemui dalam sistem ritel dan e-commerce secara praktis. Akibatnya, pertanyaan di masa depan dapat ditingkatkan dengan menggabungkan simulasi transaksi bersamaan, lingkungan database terdistribusi, koordinasi konkurensi, dan pengujian toleransi kesalahan untuk mencapai evaluasi kinerja sistem yang lebih lengkap [12], [21], [24].

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Studi ini menginvestigasi efektivitas kinerja sistem kasir dan manajemen persediaan yang dibangun menggunakan C++ dalam konteks perdagangan elektronik domestik. Metode yang digunakan adalah eksperimen komputasi dengan memanfaatkan beban kerja transaksi sintetis. Temuan dari evaluasi mengindikasikan bahwa sistem dapat menjaga kinerja yang cukup konsisten di bawah berbagai intensitas beban transaksi, mencakup skala rendah, sedang, dan tinggi. Kenaikan *volume* transaksi berdampak pada waktu eksekusi, latensi, dan konsumsi memori. Meskipun demikian, sistem berhasil mempertahankan tingkat *"throughput"* yang tinggi dan kecepatan respons transaksi yang stabil sepanjang pengujian. Penggunaan bahasa pemrograman C++ dalam implementasi ini berkontribusi secara penting pada optimalisasi waktu komputasi dan ketahanan pengelolaan memori sistem. Dengan adanya manajemen memori *low-level* yang melekat pada C++, proses untuk transaksi dan penyelarasan inventaris dapat dieksekusi dengan lebih efektif, disertai dengan pembebanan komputasi yang lebih minim saat dibandingkan dengan metode yang bergantung pada *runtime* terkelola. Temuan riset juga mengindikasikan bahwa sistem dapat mengelola peningkatan *volume* beban kerja hingga 100.000 transaksi tanpa menunjukkan penurunan kinerja yang berarti, yang menunjukkan kemampuan skalabilitas yang memadai untuk menunjang kegiatan transaksi *digital* dalam ekosistem *e-commerce* domestik.

Penelitian ini tidak hanya menyumbangkan nilai praktis dalam perancangan sistem transaksi *digital* yang lebih efisien dan andal, tetapi juga memberikan kontribusi teoritis pada analisis perbandingan kinerja sistem kasir dan inventaris yang dikembangkan menggunakan C++. Metode eksperimental yang memanfaatkan beban kerja sintetis *(synthetic workload)* memungkinkan penilaian yang lebih kuantitatif mengenai korelasi antara beban transaksi dan fluktuasi kinerja sistem. Oleh karena itu, studi ini mengindikasikan bahwa penerapan sistem transaksi berbasis C++ menawarkan prospek yang menjanjikan untuk diadopsi dalam ekosistem *e-commerce* domestik yang menuntut efisiensi memori, kapasitas pemrosesan yang tinggi, serta keandalan transaksi secara seketika. Namun demikian, studi ini menghadapi beberapa kendala, khususnya dalam penerapan beban kerja sintetis yang belum sepenuhnya mencerminkan pola transaksi di lingkungan operasional sesungguhnya. Sebagai konsekuensinya, investigasi lanjutan dapat dieksplorasi melalui pemanfaatan data transaksi riil, penerapan pemrosesan transaksi terdistribusi, sesrta pengintegrasian teknologi *cloud* dan basis data waktu nyata guna mendapatkan penilaian kinerja yang lebih mendalam dalam skala besar di lingkungan *e-commerce.*

5.2. Saran

Investigasi selanjutnya telah menunjukkan perlunya menggunakan kumpulan data transaksi otentik di

p-ISSN : 2803-1507 e-ISSN : 2803-1531

samping pembentukan protokol pengujian dalam kerangka sistem terdistribusi untuk memfasilitasi penilaian metrik kinerja yang lebih menyeluruh. Selain itu, penggabungan teknologi komputasi awan dengan sistem database *real-time* memiliki potensi signifikan untuk meningkatkan skalabilitas dan efisiensi operasional infrastruktur transaksi *digital* dalam lanskap *e-commerce modern.*

DAFTAR PUSTAKA

[1] S. Wahyuni and E. Lovita, "Implementation of Affari POS to improve system effectiveness and inventory accuracy," Journal of Public Accounting and Financial Management, vol. 5, no. 1, pp. 25–37, 2025. https://doi.org/10.36407/jpafm.v5i1.1648

[2] R. Purwasih and D. M. Candana, "Development of Inventory Management Information System in a Retail Company," Jurnal Sains Informatika Terapan, vol. 3, no. 2, 2024, doi: 10.62357/jsit.v3i2.341.

[3] L. S. Noor, D. Deriawan, and A. Oktrivina, "E-commerce and Financial Performance: Identification on Trends, Benefits and Challenges," ATESTASI: Jurnal Ilmiah Akuntansi, vol. 6, no. 1, 2023, doi: 10.57178/atestasi.v6i1.931.

[4] Agrawal, G. (2025). Scaling Retail: Understanding Large-Scale Distributed Systems in Modern E-Commerce. International Journal of Research in Computer Applications and Information Technology (IJRCAIT), 8(1), 3455–3467. https://doi.org/10.34218/IJRCAIT_08_01_248

[5] S. P. Mukkath, "Real-Time Inventory Optimization in Retail Using Streaming Data," Journal of Computer Science and Technology Studies, vol. 7, no. 3, pp. 757–765, 2025, doi: 10.32996/jcsts.2025.7.3.82.

[6] H. I. Huda, H. Kusumo, and E. Endaryati, "Optimization of Supply Chain Processes in the Retail Sector: A Data-Driven Simulation Approach for Inventory Management," JMI: Journal of Management and Informatics, vol. 3, no. 1, 2024, doi: 10.51903/jmi.v3i1.47.

[7] S. Makalalag and E. S. Soegoto, "Implementation of a Web-Based Inventory System for Sales Efficiency in Retail Sector MSMEs in Bandung City", Int. J. Res. Appl. Tech., vol. 5, no. 1, pp. 284–291, Jul. 2025, Accessed: Jun. 11, 2026. [Online]. Available: https://ojs.unikom.ac.id/index.php/injuratech/article/view/19045

[8] S. Henning and W. Hasselbring, "Benchmarking Scalability of Stream Processing Frameworks Deployed as Microservices in the Cloud," Journal of Systems and Software, vol. 208, Art. no. 111879, Feb. 2024, doi: 10.1016/j.jss.2023.111879.

[9] J. Gómez-Luna, I. E. Hajj, I. Fernandez, C. Giannoula, G. F. Oliveira, and O. Mutlu, "Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-In-Memory System," IEEE Access, vol. 10, pp. 52565–52608, 2022, doi: 10.1109/ACCESS.2022.3174101.

[10] Pagidoju, "Cost-Performance Analysis of Cloud-Based Retail Point-of-Sale Systems: A Comparative Study of Google Cloud Platform and Microsoft Azure," In: Roy, I., Kambhampaty, K.K., Goto, T., Hu, G. (eds) Computer Applications in Industry and Engineering. CAINE 2025. Communications in Computer and Information Science, vol 2740. Springer, 2026, Cham. https://doi.org/10.1007/978-3-032-14893-3_7

[11] R. Ramdhani, A. Sujjada, and N. Nugraha, "Enhancing E-Commerce System Scalability Through Event-Driven Architecture with RabbitMQ and Docker", bit-Tech, vol. 8, no. 2, pp. 1437–1445, Dec. 2025.

[12] S. Henning and W. Hasselbring, "A Configurable Method for Benchmarking Scalability of Cloud-Native Applications," Empirical Software Engineering, vol. 27, no. 6, Art. no. 143, 2022, doi: 10.1007/s10664-022-10162-1.

[13] M. I. Ali, R. D. Fardiarsyah, L. Shodik, F. Z. D. Kinanti, and I. P. Pujiono, "Analisis Komparatif Efisiensi Memori dan Waktu Komputasi pada 8 Algoritma Sorting menggunakan C++," LogicLink: Journal of Artificial Intelligence and Multimedia in Informatics, vol. 2, no. 1, pp. 1–17, Jun. 2025, doi: 10.28918/logiclink.v2i1.10868.

[14] D. A. Saputra, M. S. Ramadhani, M. A. Failandri, A. Turmudi, and I. P. Pujiono, "Analisis Perbandingan Algoritma Sorting dalam Javascript untuk Meningkatkan Efisiensi Pengurutan Produk pada Aplikasi E-Commerce," Sainstech: Jurnal Penelitian dan Pengkajian Sains dan Teknologi, vol. 35, no. 2, pp. 1–10, Jun. 2025, doi: 10.37277/stch.v35i2.2337.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

[15] L. Qu, Y. Li, R. Zhang, T. Chen, K. Shu, and W. Qian, "Application-Oriented Workload Generation for Transactional Database Performance Evaluation," in 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 2022, doi: 10.1109/ICDE53745.2022.00036.

[16] S. N. Okofu, C. Asuai, O. Okumoku-Evroro, and A. Maureen, "Development of an Enhanced Point of Sales System for Retail Business in Developing Countries," Business and Humanistic Insights, vol. 11, no. 1, Mar. 2025, doi: 10.22624/AIMS/BHI/V11N1P1.

[17] D. Y. Prasetyo and F. Yunita, "Sistem Informasi E-Kasir pada Berry Konveksi Tembilahan," Riset dan E-Jurnal Manajemen Informatika Komputer (REMiK), vol. 6, no. 4, pp. 875–885, Oct. 2022, doi: 10.33395/remik.v6i4.11873.

[18] R. Kumar, "Event-Driven Architectures for Real-Time Data Processing: A Deep Dive Into System Design and Optimization," Mar. 2023, doi: 10.5281/zenodo.15026990.

[19] R. Yulia and D. N. D. Seputro, "Digital Transformation of Inventory Management: Evaluating the Impact of First in First Out-Based Accurate Point of Sale System Implementation on Warehouse Operations," International Journal of Accounting, Business and Organization (IJABO), vol. 6, no. 3, pp. 378–393, Dec. 2025, doi: 10.61242/ijabo.25.663.

[20] C. Zhang, Y. Li, R. Zhang, W. Qian, and A. Zhou, "Benchmarking for Transaction Processing Database Systems in Big Data Era," in Benchmarking, Measuring, and Optimizing, Lecture Notes in Computer Science, Cham, Switzerland: Springer, 2019, pp. 147–158, doi: 10.1007/978-3-030-32813-9_13.

[21] J. Li, Y. Lu, Y. Zhang, Q. Wang, Z. Cheng, K. Huang, and J. Shu, "Switchtx: Scalable In-Network Coordination for Distributed Transaction Processing," Proc. VLDB Endow., vol. 15, no. 11, pp. 2881–2894, Jul. 2022, doi: 10.14778/3551793.3551838.

[22] R. K. Chennareddy, "Designing Data and Analytics Ecosystems for High Volume Transaction Processing Applications," International Journal of Artificial Intelligence, Big Data Computing and Data Management Systems (IJAIBDCMS), vol. 2, no. 2, pp. 95–106, Jun. 2021, doi: 10.63282/3050-9416.IJAIBDCMS-V2I2P111.

[23] H. Hu, X. Zhou, T. Zhu, W. Qian, and A. Zhou, "In-Memory Transaction Processing: Efficiency and Scalability Considerations," Knowledge and Information Systems, vol. 61, no. 3, pp. 1209–1240, Dec. 2019, doi: 10.1007/s10115-019-01340-7.

[24] J. Hildred, M. Abebe, and K. Daudjee, "Caerus: Low-Latency Distributed Transactions for Geo-Replicated Systems," Proceedings of the VLDB Endowment, vol. 17, no. 3, pp. 469–482, Nov. 2023, doi: 10.14778/3632093.3632109.

[25] V. B. Sarabu, "Scalable Data Processing Patterns for National Retail Platforms: An Enterprise Architecture for High-Volume Transaction Systems," International Journal of Computer Technology and Engineering Computing (IJCTEC), vol. 3, no. 3, pp. 1–14, Jun. 2020, doi: 10.15680/f1jhg344.

[26] P. Chatterjee, "Real-Time Payment System and Their Scalability Challenges," Iconic Research and Engineering Journals, vol. 6, no. 12, pp. 1461–1470, Jun. 2023, doi: 10.2139/ssrn.5101064.