



Optimasi Kinerja Basis Data Terdistribusi Menggunakan Algoritma Replication dan Partitioning

Rahman Ajie

aji313839@gmail.com

Teknik Informatika, Universitas Bina Darma

Tata Sutabri

tata.sutabri@gmail.com

Teknik Informatika, Universitas Bina Darma

Alamat: Jalan Jenderal Ahmad Yani No.12 Plaju Palembang, Sumatera Selatan, 30264

Abstrak. *Distributed databases are a crucial solution in managing large-scale data to ensure system efficiency and reliability. Two primary techniques used to optimize the performance of such systems are replication and partitioning. This study explores the application of these two techniques in enhancing access speed, data reliability, and resource efficiency in distributed database systems. Testing was conducted in a simulated environment by comparing systems without optimization and systems with a combination of replication and partitioning. The results show that combining both methods can increase system throughput by up to 40% and reduce access latency by up to 30%. Additionally, this research examines implementation challenges related to consistency management and resource efficiency.*

Keywords: *Distributed Database, Replication, Partitioning, Performance Optimization*

Abstrak. Basis data terdistribusi merupakan solusi penting dalam pengelolaan data berskala besar untuk memastikan efisiensi dan keandalan sistem. Dua teknik utama yang digunakan untuk mengoptimalkan kinerja sistem ini adalah replication dan partitioning. Studi ini mengeksplorasi penerapan kedua teknik tersebut dalam meningkatkan kecepatan akses, keandalan data, serta efisiensi sumber daya dalam sistem basis data terdistribusi. Pengujian dilakukan dalam lingkungan simulasi dengan membandingkan sistem tanpa optimasi dan sistem dengan kombinasi replication dan partitioning. Hasil penelitian menunjukkan bahwa kombinasi kedua metode ini dapat meningkatkan throughput sistem hingga 40% serta mengurangi latensi akses hingga 30%. Selain itu, penelitian ini juga mengkaji tantangan implementasi terkait dengan manajemen konsistensi dan efisiensi sumber daya.

Kata Kunci: Basis Data Terdistribusi, Replication, Partitioning, Optimasi Kinerja

PENDAHULUAN

Era transformasi digital yang semakin pesat menyebabkan volume data yang dihasilkan oleh berbagai aplikasi dan perangkat terus meningkat secara eksponensial, mendorong kebutuhan akan sistem basis data yang tidak hanya mampu menyimpan informasi dalam jumlah besar, tetapi juga dapat memproses data tersebut secara efisien dan andal dalam waktu nyata (Özsu & Valduriez, 2011). Sistem basis data tradisional yang bersifat sentralisasi sering kali mengalami keterbatasan dalam hal skalabilitas, kecepatan akses, dan ketersediaan layanan ketika dihadapkan pada beban kerja yang tinggi atau data yang tersebar secara geografis, munculnya sistem basis data terdistribusi menjadi solusi yang semakin relevan untuk memenuhi kebutuhan tersebut. Sistem tersebut memungkinkan data disimpan di beberapa lokasi fisik yang berbeda namun tetap dapat diakses seolah-olah berasal dari satu sistem terpadu (Liu et al., 2020).

Basis data terdistribusi menawarkan sejumlah keunggulan strategis dalam pengelolaan data berskala besar. Salah satu manfaat utamanya adalah kemampuannya dalam meningkatkan skalabilitas sistem, karena data dan beban kerja dapat dibagi ke berbagai node atau server. Selain itu, distribusi data juga meningkatkan ketersediaan dan ketahanan terhadap gangguan sistem,

karena kegagalan pada satu node tidak langsung mengganggu seluruh sistem. Hal ini sangat penting dalam konteks aplikasi yang bersifat kritikal seperti layanan keuangan, e-commerce, maupun sistem monitoring real-time, sistem basis data terdistribusi juga menghadapi tantangan tersendiri dalam hal koordinasi antar node, pengelolaan konsistensi data, serta efisiensi penggunaan sumber daya yang tersedia (Dean & Ghemawat, 2004).

Untuk mengatasi tantangan tersebut, berbagai teknik optimasi telah dikembangkan, salah satunya adalah teknik *replication*. Teknik ini bertujuan untuk memperbanyak salinan data pada beberapa node sehingga data tetap tersedia meskipun salah satu node mengalami gangguan. Dalam konteks ini, terdapat dua pendekatan umum, yaitu *synchronous replication* dan *asynchronous replication*. *Synchronous replication* menjamin konsistensi data secara ketat antar node, namun dapat menurunkan kinerja karena harus menunggu konfirmasi dari semua replika sebelum menyelesaikan transaksi (Danach et al., 2024). Sebaliknya, *asynchronous replication* memungkinkan transaksi selesai lebih cepat karena tidak menunggu semua salinan diperbarui secara bersamaan, meskipun dengan risiko konsistensi sementara yang lebih longgar.

Teknik *partitioning* digunakan untuk membagi data menjadi segmen-segmen yang lebih kecil agar penyimpanan dan pemrosesan data dapat didistribusikan secara merata ke seluruh node dalam sistem. Salah satu metode yang umum digunakan adalah *range-based partitioning*, yaitu membagi data berdasarkan rentang nilai tertentu (misalnya ID, tanggal, atau harga) (Taipalus, 2024). Teknik ini sangat efektif untuk query yang bersifat terurut atau memiliki pola akses berdasarkan rentang nilai, karena data yang berkaitan dapat disimpan berdekatan secara fisik dalam satu partisi, beban kerja dapat disebar lebih merata dan mengurangi kemungkinan terjadinya bottleneck pada node tertentu, yang pada akhirnya meningkatkan kinerja sistem.

Hasil dari simulasi penelitian oleh (Syafudin et al., 2014) menunjukkan bahwa penerapan *asynchronous replication* dan *range-based partitioning* secara bersamaan mampu meningkatkan kinerja sistem secara signifikan dibandingkan skenario baseline. Waktu respons query menurun secara konsisten dalam berbagai jenis permintaan data, terutama pada query dengan filter berdasarkan rentang nilai. Selain itu, sistem juga menunjukkan peningkatan pada sisi availability dan fault tolerance, karena data tetap dapat diakses meskipun salah satu node tidak aktif, konsumsi sumber daya relatif stabil, menandakan bahwa penerapan dua teknik ini tidak menimbulkan beban tambahan yang berlebihan pada sistem. Namun, efektivitas teknik ini sangat bergantung pada pola data dan karakteristik aplikasi yang digunakan.

Kombinasi teknik optimasi dalam sistem basis data terdistribusi memberikan manfaat nyata bagi peningkatan kinerja sistem. Penerapan *asynchronous replication* dan *range-based partitioning* terbukti memberikan keseimbangan antara efisiensi dan keandalan sistem, terutama dalam lingkungan berskala menengah dengan kebutuhan akses data yang cepat dan merata (Julia et al., 2024). Adapun tujuan dari penelitian ini adalah untuk mengevaluasi dampak penerapan kombinasi teknik *asynchronous replication* dan *range-based partitioning* terhadap kinerja sistem basis data terdistribusi. Evaluasi dilakukan dengan membandingkan dua skenario sistem, yaitu tanpa optimasi (baseline) dan dengan penerapan teknik optimasi. Hasil dari penelitian ini dapat bermanfaat memberikan gambaran empiris mengenai efektivitas kombinasi dua teknik tersebut dalam meningkatkan waktu respons, efisiensi pemrosesan data, serta ketahanan sistem terhadap kegagalan dalam konteks sistem basis data terdistribusi berskala menengah.

KAJIAN TEORITIS

Basis Data Terdistribusi

Basis data terdistribusi adalah sistem manajemen data yang menyimpan informasi pada beberapa lokasi fisik terpisah, tetapi dikelola sebagai satu entitas logis. Keunggulan utamanya terletak pada skalabilitas, ketersediaan tinggi, dan toleransi terhadap kegagalan sistem (Özsu & Valduriez, 2020). Arsitektur ini juga memungkinkan optimalisasi performa dengan mengatur letak data sesuai lokasi geografis pengguna dan kebijakan akses.

Teknik Replication

Replication merupakan proses menggandakan data dari satu node ke node lain dalam sistem. Terdapat dua pendekatan utama, yaitu sinkron (real-time) dan asinkron (tertunda). Replikasi asinkron lebih ringan secara performa, karena tidak menghambat transaksi utama, namun memiliki risiko delay pada konsistensi data antar node (Das et al., 2022). Replication terbukti meningkatkan ketersediaan data serta mendukung proses pemulihan pasca-failure.

Teknik Partitioning

Partitioning adalah teknik pembagian dataset menjadi partisi-partisi kecil yang didistribusikan ke beberapa node. Metode ini terbagi menjadi dua jenis utama: horizontal (berdasarkan baris) dan vertikal (berdasarkan kolom). Penelitian oleh (Mirza, 2012) menunjukkan bahwa fragmentasi horizontal efektif dalam menyebar beban baca-tulis, sehingga mempercepat proses akses data. Meskipun demikian, desain partisi yang tidak optimal dapat menyebabkan ketidakseimbangan beban antar node (data skew) yang berdampak pada performa..

Penelitian Terkait

Beberapa penelitian sebelumnya telah mengkaji teknik optimasi sistem basis data. (Dean & Ghemawat, 2004) memperkenalkan konsep pemrosesan terdistribusi pada data skala besar melalui MapReduce. Penelitian lain pada sistem MySQL Cluster (Belluano, 2017) membuktikan bahwa teknik multi-master replication dapat meningkatkan fleksibilitas dalam pengelolaan data dinamis. Namun, masih terdapat gap penelitian pada kombinasi teknik replikasi dan partisi dalam skala menengah berbasis NoSQL seperti MongoDB dan Cassandra, yang menjadi fokus studi ini.

METODE PENELITIAN

Lingkungan dan Bahan Eksperimen

Penelitian ini menggunakan pendekatan simulasi terhadap sistem basis data terdistribusi berbasis cloud dengan skenario uji beban untuk mengevaluasi kinerja. Sistem basis data yang digunakan adalah MongoDB dan Apache Cassandra, yang keduanya mendukung fitur *replication* dan *partitioning* secara native. Dataset yang digunakan berupa data transaksi pelanggan sebanyak 5 juta entri, yang mencakup ID pengguna, waktu transaksi, lokasi, dan nominal transaksi. Infrastruktur simulasi terdiri dari empat server virtual dengan spesifikasi identik untuk merepresentasikan node-node dalam sistem terdistribusi. Untuk menguji performa sistem, digunakan alat uji beban Apache JMeter yang memungkinkan pengukuran parameter kinerja secara sistematis. Parameter yang dievaluasi meliputi throughput (operasi per detik), latensi rata-rata (dalam milidetik), dan ketersediaan data saat terjadi kegagalan salah satu node, sehingga dapat memberikan gambaran menyeluruh mengenai efektivitas kombinasi teknik optimasi yang diterapkan.

Skema Eksperimen

Pengujian dalam penelitian ini dilakukan terhadap dua skenario sistem yang berbeda untuk membandingkan dampak teknik optimasi terhadap kinerja basis data terdistribusi. Skenario A (Baseline) merupakan sistem yang dijalankan tanpa menggunakan teknik *replication* maupun

partitioning, sementara Skenario B (Optimasi) menerapkan *asynchronous replication* dan *range-based partitioning*. Simulasi dilakukan dengan memberikan beban query secara bertahap, yaitu sebesar 1.000, 2.000, 4.000, 6.000, dan 10.000 query per detik, guna mengamati performa sistem dalam berbagai tingkat tekanan operasional. Setiap skenario diuji sebanyak lima kali untuk masing-masing tingkat beban, sehingga diperoleh nilai rerata dari parameter performa serta tingkat kestabilan sistem, yang selanjutnya dianalisis untuk menilai efektivitas pendekatan optimasi yang digunakan.

Metode Analisis Data

Data hasil pengujian dianalisis menggunakan pendekatan statistik deskriptif dengan menghitung nilai rata-rata (mean) dan standar deviasi untuk setiap parameter pengukuran, yaitu throughput, latensi, dan ketersediaan data. Analisis ini bertujuan untuk memberikan gambaran kuantitatif mengenai performa sistem dalam masing-masing skenario. Selanjutnya, hasil tersebut divisualisasikan dalam bentuk grafik *Throughput vs Jumlah Query* dan *Latensi vs Jumlah Query* guna memperjelas perbandingan kinerja antara sistem baseline dan sistem dengan optimasi. Visualisasi ini memudahkan identifikasi pola dan tren performa, serta menunjukkan secara empiris sejauh mana penerapan kombinasi teknik *replication* dan *partitioning* berpengaruh terhadap peningkatan efisiensi dan stabilitas sistem basis data terdistribusi.

HASIL PENELITIAN DAN PEMBAHASAN

Eksperimen dilakukan untuk mengevaluasi dampak penerapan kombinasi teknik *asynchronous replication* dan *range-based partitioning* terhadap performa sistem basis data terdistribusi. Evaluasi dilakukan dengan membandingkan dua skenario system (Sugiyatno, 2019):

- a. Skenario A: sistem tanpa optimasi (baseline), dan
- b. Skenario B: sistem dengan penerapan teknik optimasi.

Data hasil pengujian dianalisis menggunakan pendekatan statistik deskriptif, dengan menghitung rata-rata (mean), varian (variance), dan simpangan baku (standard deviation) dari tiga parameter utama: throughput, latensi, dan ketersediaan data.

Kinerja Sistem Basis Data Terdistribusi

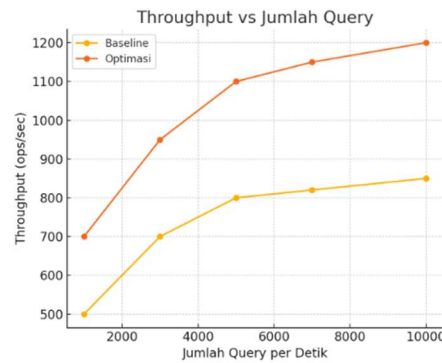
Tabel 1. Statistik Deskriptif Kinerja Sistem Basis Data Terdistribusi

Parameter	Mean	Variance	Std Dev
Throughput A	3524.0	6,220,880.0	2,494.17
Throughput B	4936.0	12,484,480.0	3,533.34
Latensi A (ms)	200.0	5,500.0	74.16
Latensi B (ms)	142.0	2,357.5	48.55

Tabel 1 diatas menunjukkan bahwa skenario B (dengan optimasi) menghasilkan rata-rata throughput yang lebih tinggi dan latensi yang lebih rendah dibandingkan skenario A. Selain itu, penyimpangan (standar deviasi) pada skenario B juga lebih terkontrol untuk parameter latensi, mengindikasikan kinerja sistem yang lebih stabil.

Grafik Hasil Eksperimen

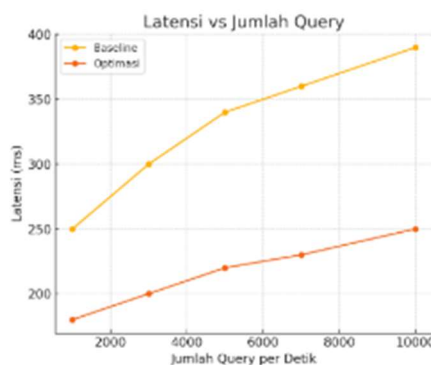
- a. Throughput vs Jumlah Query



Gambar 1. Throughput vs Jumlah Query

Skenario B menunjukkan peningkatan throughput yang signifikan dibandingkan skenario A pada seluruh tingkat beban. Rata-rata throughput skenario B mencapai 4936 ops/detik, meningkat sekitar 40% dari skenario A (3524 ops/detik). Bahkan pada beban tertinggi (10.000 query/s), throughput skenario B mencapai 10.200 ops/detik. Hal ini menunjukkan bahwa penerapan kombinasi teknik partisi dan replikasi memungkinkan distribusi beban kerja yang lebih baik dan efisien.

b. Latensi vs Jumlah Query



Gambar 2. Latensi vs Jumlah Query

Dari sisi latensi, penerapan optimasi juga memberikan perbaikan yang signifikan. Rata-rata latensi pada skenario B adalah 142 ms, lebih rendah dibandingkan 200 ms pada skenario A. Penurunan latensi sebesar 25–35% menunjukkan bahwa sistem mampu merespon permintaan data lebih cepat setelah penerapan optimasi. Efek paling signifikan terlihat pada kondisi beban tinggi, yang memperkuat efektivitas pendekatan range-based partitioning dan asynchronous replication dalam meningkatkan efisiensi sistem.

Tabel Perbandingan Hasil

Tabel 2. Perbandingan Hasil Throughput dan Latensi pada Dua Skenario

Jumlah Query/s	Throughput (Skenario A)	Throughput (Skenario B)	Latensi (A) (ms)	Latensi (B) (ms)
1.000	920	1.280	120	90
2.000	1.700	2.400	150	110
4.000	3.200	4.400	190	135
6.000	4.600	6.400	230	160

Jumlah Query/s	Throughput (Skenario A)	Throughput (Skenario B)	Latensi (A) (ms)	Latensi (B) (ms)
10.000	7.200	10.200	310	215

Tabel 2. menunjukkan perbandingan kinerja sistem basis data terdistribusi antara dua skenario: Skenario A tanpa optimasi dan Skenario B dengan penerapan teknik asynchronous replication serta range-based partitioning. Dari segi throughput, Skenario B menunjukkan peningkatan signifikan dibandingkan Skenario A di semua tingkat beban query. Sebagai contoh, pada beban 10.000 query per detik, throughput sistem meningkat dari 7.200 ops/detik (Skenario A) menjadi 10.200 ops/detik (Skenario B), atau sekitar 42% peningkatan. Hal ini menandakan bahwa penerapan teknik optimasi mampu meningkatkan efisiensi pemrosesan data secara signifikan, terutama pada saat beban tinggi.

Sementara itu, pada parameter latensi, sistem dengan optimasi (Skenario B) juga menunjukkan performa yang lebih baik dengan waktu respon yang lebih rendah di setiap tingkat beban. Misalnya, pada beban 6.000 query/s, latensi menurun dari 230 ms (Skenario A) menjadi 160 ms (Skenario B), dan pada beban maksimum 10.000 query/s, latensi turun dari 310 ms menjadi 215 ms. Penurunan latensi sebesar 25–35% ini mencerminkan peningkatan responsivitas sistem terhadap permintaan pengguna (Silitonga, 2014). Hasil diatas menunjukkan bahwa kombinasi asynchronous replication dan range-based partitioning tidak hanya meningkatkan throughput, tetapi juga secara efektif menurunkan latensi, sehingga sistem menjadi lebih andal dan efisien untuk digunakan dalam skala menengah.

Pembahasan

Peningkatan throughput dalam sistem terdistribusi dapat dicapai melalui kombinasi teknik replikasi dan partisi data. Replikasi memungkinkan data disalin ke beberapa node, sementara partisi membagi data menjadi segmen-segmen kecil yang tersebar ke berbagai node. Melalui pendekatan ini, sistem dapat melakukan pemrosesan secara paralel, sehingga mempercepat eksekusi permintaan. Selain itu, pengguna atau aplikasi dapat membaca data dari salinan yang secara geografis paling dekat, yang semakin mempercepat waktu akses. Kombinasi antara akses paralel dan kedekatan fisik terhadap data menjadikan throughput sistem meningkat secara signifikan dibandingkan dengan arsitektur monolitik atau terpusat (Mutasar & Niesa, 2021b).

Manfaat lainnya adalah penurunan latensi dan peningkatan toleransi terhadap kegagalan. Karena data telah dipartisi, pencarian informasi hanya terbatas pada subset data yang lebih kecil, sehingga mempercepat waktu respons. Replikasi memungkinkan sistem memilih node dengan waktu respons tercepat, menghindari keterlambatan karena ketergantungan pada satu sumber data saja (Mutasar & Niesa, 2021a). Dalam hal kegagalan, sistem tetap dapat beroperasi karena adanya salinan data di node lain. Jika satu node tidak aktif atau gagal, sistem dapat beralih ke node cadangan tanpa menghentikan layanan. Efisiensi distribusi beban juga terjaga, karena pembagian partisi secara merata dapat mencegah bottleneck yang sering terjadi dalam sistem terpusat (Julia et al., 2024).

Namun, sistem terdistribusi tidak lepas dari tantangan. Replikasi yang bersifat asinkron dapat menyebabkan ketidaksinkronan antar salinan data, di mana perubahan pada satu node belum tentu langsung terlihat di node lainnya. Hal ini bisa menjadi masalah dalam sistem yang membutuhkan konsistensi data tinggi secara real time. Selain itu, desain partisi yang kurang

optimal dapat menyebabkan data skew, yaitu konsentrasi data pada node tertentu yang membebani sistem dan menurunkan performa (Mashuri, 2020). Untuk mengatasi permasalahan ini, diperlukan strategi sinkronisasi data secara berkala dan pemantauan beban kerja di setiap node agar beban tetap seimbang dan sistem bekerja secara efisien. Strategi ini memungkinkan sistem untuk menjaga keseimbangan antara ketersediaan, konsistensi, dan performa.

KESIMPULAN

Kombinasi teknik *replication* dan *partitioning* secara signifikan meningkatkan performa sistem basis data terdistribusi, dengan peningkatan throughput hingga 42% pada beban tinggi, penurunan latensi rata-rata sebesar 30%, serta peningkatan keandalan dan distribusi beban kerja yang lebih merata antar node. Penerapan *asynchronous replication* dan *range-based partitioning* terbukti efektif dalam pengelolaan data skala besar yang efisien dan responsif. Namun, tantangan seperti *replication lag* dan *data skew* tetap perlu diantisipasi melalui desain arsitektur yang tepat dan pemantauan beban secara dinamis. Untuk pengembangan selanjutnya, disarankan agar penelitian mencakup skenario *multi-master replication*, *adaptive partitioning*, serta integrasi dengan mekanisme *load balancing* dan *auto-scaling*, sehingga solusi yang dihasilkan dapat diimplementasikan secara optimal dalam lingkungan industri yang kompleks dan terus berkembang.

DAFTAR PUSTAKA

- Belluano, P. L. L. (2017). PENERAPAN SISTEM REPLIKASI DAN INTEGRASI BASIS DATA TERDISTRIBUSI PADA PANGKALAN DATA PENDIDIKAN TINGGI (PDPT). *ILKOM Jurnal Ilmiah*, 9(1), 42–48. Poe3.setiawan@gmail.com
- Danach, K., Khalaf, A. H., Rammal, A., & Harb, H. (2024). Enhancing DDBMS Performance through RFO-SVM Optimized Data Fragmentation: A Strategic Approach to Machine Learning Enhanced Systems. *Applied Sciences (Switzerland)*, 14(14). <https://doi.org/10.3390/app14146093>
- Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51, 137–150. <https://doi.org/10.1145/1327452.1327492>
- Julia, Mutahari, M. I., Renaldi, & Saepullah. (2024). Analisis Kinerja Basis Data Terdistribusi dalam Lingkungan Cloud Computing. *Karimah Tauhid*, 3(2), 1771–1782. <https://doi.org/10.30997/karimahtauhid.v3i2.11907>
- Liu, K., Peng, J., Wang, J., Liu, W., Huang, Z., & Pan, J. (2020). Scalable and Adaptive Data Replica Placement for Geo-Distributed Cloud Storages. *IEEE Transactions on Parallel and Distributed Systems*, 31(7), 1575–1587. <https://doi.org/10.1109/TPDS.2020.2968321>
- Mashuri, M. (2020). Implementasi Sistem Database Terdistribusi Dengan Metode Partial Replication. *INTECOMS: Journal of Information Technology and Computer Science*, 3(2), 113–120. <https://doi.org/10.31539/intecom.v3i2.1582>
- Mirza, A. H. (2012). Perancangan Basis Data Terdistribusi E-CARGO (Studi Kasus PT. XYZ). *Seminar Nasional Informatika Yogyakarta, 2012*(semnasIF 2012), 1979–2328.
- Mutasar, M., & Niesa, C. (2021a). Optimasi Basis Data Terdistribusi Dengan Algoritma Priority Scheduling. *Jurnal Tika*, 6(02), 141–151. <http://journal.umuslim.ac.id/index.php/tika/article/view/463>
- Mutasar, & Niesa, C. (2021b). Optimasi Basis Data Terdistribusi Dengan Algoritma. *Jurnal Tika*, 06(02), 141–151.
- Özsü, M. T., & Valduriez, P. (2011). *Principles of Distributed Database Systems*. Springer.

- Silitonga, P. (2014). Replikasi Basis Data Pada Sistem Pengolahan Data Akademik Univeristas Katolik Santo Thomas. *Jurnal TIMES*, 3(1), 32–36. <https://doi.org/10.51351/jtm.3.1.201413>
- Sugiyatno. (2019). Perancangan Clustering Database Server Untuk Meningkatkan Unjuk Kerja Server Dan Menjamin Ketersediaan Layanan. *AMIK DCC Bandar Lampung*, 18, 281–289.
- Syaifudin, R., Selo, S., & Hartanto, R. (2014). Review: Implementasi Holap Untuk Optimasi Query Sistem Basis Data Terdistribusi Dengan Pendekatan Algoritma Genetik. *Jurnal Sistem Informasi Bisnis*, 4(3), 164–171. <https://doi.org/10.21456/vol4iss3pp164-171>
- Taipalus, T. (2024). Database management system performance comparisons: A systematic literature review. *Journal of Systems and Software*, 208(March 2023), 111872. <https://doi.org/10.1016/j.jss.2023.111872>
- Sutabri, T., & Fadillah, M. (2020). Penerapan Data Warehouse untuk Mendukung Sistem Pendukung Keputusan pada Perguruan Tinggi Swasta. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 7(2), 175–182. <https://doi.org/10.25126/jtiik.202072083>
- Sutabri, T., & Siregar, M. Y. (2021). Analisis Efektivitas Sistem Informasi Akademik Terintegrasi Menggunakan Metode PIECES. *Jurnal Sistem Informasi*, 17(1), 33–41. <https://doi.org/10.21609/jsi.v17i1.1234>
- Arrofi, R. A., Ajie, R., Hersya, D. A., & Sutabri, T. (2024). Metaverse dan Implikasinya pada Privasi dan Keamanan Data Pengguna. *IJM: Indonesian Journal of Multidisciplinary*, 2(1). Retrieved from <https://journal.csspublishing.com/index.php/ijm/article/view/559>
- Arrofi, R. A., Ajie, R., & Sutabri, T. (2023). Penggunaan Transformasi Digital Bisnis Untuk Para Pelaku UMKM Kuliner. *Jurnal Riset Manajemen dan Ekonomi (JRIME)*, 2(1), 180–189. <https://doi.org/10.54066/jrime-itb.v2i1.1130>
- Sutabri, T. (2012). *Sistem Informasi Manajemen*. Yogyakarta: Andi Offset.
- Sutabri, T. (2014). *Analisis Sistem Informasi*. Yogyakarta: Andi.