

Analisis kinerja database terdistribusi pada jaringan Software-Defined Networking (SDN) untuk optimalisasi sistem informasi akademik

Septa Petriansyah¹, Fabio Junior¹, R Wahyudi Darmawan¹

¹Universitas Tangerang Raya, Kabupaten Tangerang, Banten

Penulis Korespondensi : Septa Petriansyah (e-mail: septapetriansyah@gmail.com)

ABSTRAK

Perkembangan sistem informasi akademik di perguruan tinggi memerlukan infrastruktur yang scalable dan reliable untuk menangani concurrent access yang tinggi, terutama saat periode kritis seperti pendaftaran KRS. Penelitian ini menganalisis kinerja database terdistribusi pada jaringan Software-Defined Networking untuk optimalisasi sistem informasi akademik di STIT Muhammadiyah Banjar. Metode eksperimental digunakan dengan membandingkan empat skenario: database terpusat pada jaringan tradisional, database terpusat pada jaringan SDN, database terdistribusi pada jaringan tradisional, dan database terdistribusi pada jaringan SDN. Pengujian dilakukan dengan variasi workload 100 hingga 5000 concurrent users menggunakan PostgreSQL dengan Citus extension dan ONOS controller. Hasil penelitian menunjukkan kombinasi database terdistribusi dengan SDN menghasilkan performa terbaik dengan query response time 521.4 ms pada 5000 users, 71.8% lebih cepat dibanding baseline, throughput mencapai 9589 TPS, error rate di bawah 2%, dan system availability 99.82%. Implementasi SDN menurunkan network latency hingga 43.8% dan packet loss rate hingga 79.3%. Uji Kruskal-Wallis menunjukkan perbedaan signifikan antar skenario dengan effect size besar. Penelitian ini memberikan blueprint teknis bagi institusi pendidikan tinggi dalam merancang infrastruktur IT yang optimal dengan total cost of ownership 5 tahun yang lebih efisien dibanding solusi terpusat.

KATA KUNCI : database terdistribusi; Software-Defined Networking; sistem informasi akademik; PostgreSQL Citus; ONOS controller.

1. PENDAHULUAN

Perkembangan teknologi informasi di era digital telah mengubah cara institusi pendidikan tinggi mengelola data dan informasi akademik. Secara global, transformasi digital dalam pendidikan tinggi mengalami akselerasi signifikan, terutama pasca pandemi COVID-19 yang memaksa institusi untuk mengadopsi sistem informasi berbasis cloud dan terdistribusi [1]. Menurut laporan Educause (2023), lebih dari 78% institusi pendidikan tinggi di seluruh dunia telah mengimplementasikan sistem manajemen database terpusat untuk mendukung operasional akademik, namun menghadapi tantangan signifikan terkait skalabilitas dan performa jaringan saat menghadapi lonjakan traffic pengguna [2].

Database terdistribusi telah menjadi solusi populer untuk mengatasi keterbatasan sistem database terpusat, khususnya dalam menangani big data dan concurrent access [3]. Arsitektur database terdistribusi memungkinkan data disimpan di multiple nodes yang saling terhubung melalui jaringan, sehingga meningkatkan availability, fault tolerance, dan

scalability sistem [4]. Namun, performa database terdistribusi sangat bergantung pada infrastruktur jaringan yang mendasarinya, terutama terkait dengan latency, bandwidth, dan reliability koneksi antar node.

Di sisi lain, Software-Defined Networking (SDN) telah muncul sebagai paradigma baru dalam arsitektur jaringan yang memisahkan control plane dari data plane, memungkinkan manajemen jaringan yang lebih fleksibel dan programmable. SDN memberikan kemampuan untuk mengoptimalkan routing, load balancing, dan quality of service (QoS) secara dinamis melalui controller terpusat [5]. Penelitian menunjukkan bahwa implementasi SDN dapat meningkatkan performa aplikasi terdistribusi hingga 35% dibandingkan dengan arsitektur jaringan tradisional [6].

Beberapa penelitian telah mengeksplorasi optimalisasi database terdistribusi dan implementasi SDN secara terpisah. Zhang et al. (2021) meneliti optimalisasi query processing dalam database terdistribusi menggunakan machine learning, namun tidak mempertimbangkan aspek infrastruktur jaringan [7]. Sementara itu, implementasi SDN untuk

meningkatkan QoS pada jaringan kampus, tetapi tidak mengintegrasikan analisis dengan sistem database. Hanya sedikit penelitian yang mengkombinasikan kedua teknologi ini secara komprehensif, terutama dalam konteks sistem informasi akademik di institusi pendidikan tinggi [8], [9].

Research gap yang teridentifikasi adalah minimnya penelitian empiris yang menganalisis secara holistik bagaimana arsitektur jaringan SDN dapat mengoptimalkan kinerja database terdistribusi, khususnya untuk use case sistem informasi akademik yang memiliki karakteristik unik seperti bursty traffic pattern saat periode pendaftaran, pola akses read-heavy, dan kebutuhan konsistensi data yang tinggi. Mayoritas penelitian existing fokus pada environment enterprise atau cloud computing skala besar, bukan pada konteks institusi pendidikan dengan keterbatasan resources dan infrastruktur [10], [11].

STIT Muhammadiyah Banjar, sebagai institusi pendidikan tinggi Islam yang berkembang pesat, menghadapi tantangan dalam mengelola sistem informasi akademik yang reliable dan scalable. Dengan jumlah mahasiswa yang terus meningkat dan diversifikasi program penelitian, institusi mengalami bottleneck pada sistem database saat peak hours, seperti periode registrasi online dan pengisian Kartu Rencana Penelitian (KRS). Permasalahan ini mengakibatkan slow response time, timeout connections, dan user experience yang buruk (observasi awal, 2025).

Penelitian ini penting dilakukan karena dapat memberikan blueprint teknis bagi institusi pendidikan tinggi, khususnya Perguruan Tinggi Keagamaan Islam (PTKI), dalam merancang infrastruktur IT yang optimal dengan memanfaatkan teknologi database terdistribusi dan SDN [12]. Penelitian ini menghasilkan best practices dan performance benchmarks yang dapat menjadi referensi bagi institusi sejenis dalam melakukan digital transformation. Berdasarkan perspektif akademis, penelitian ini memperkaya body of knowledge tentang integrasi teknologi database dan networking dalam konteks specific domain yaitu pendidikan tinggi Islam di Indonesia. Penelitian ini sejalan dengan Rencana Induk Riset Nasional (RIRN) 2020-2045 yang menekankan pentingnya penelitian di bidang teknologi informasi untuk mendukung transformasi digital sektor pendidikan (Kemenristekdikti, 2020). Optimalisasi infrastruktur IT di institusi pendidikan tinggi tidak hanya meningkatkan efisiensi operasional, tetapi juga berkontribusi pada peningkatan kualitas layanan pendidikan dan daya saing institusi di era Society 5.0 [2].

2. BAHAN DAN METODE

2.1. DESAIN PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimental untuk menganalisis kinerja database terdistribusi pada jaringan Software-Defined Networking (SDN). Desain penelitian yang digunakan adalah experimental design with comparison group, di mana dilakukan perbandingan performa antara empat skenario implementasi: (1) database terpusat pada jaringan tradisional sebagai baseline, (2) database

terpusat pada jaringan SDN, (3) database terdistribusi pada jaringan tradisional, dan (4) database terdistribusi pada jaringan SDN sebagai skenario yang diusulkan [1]. Penelitian dilaksanakan di STIT Muhammadiyah Banjar, Kalimantan Selatan, selama 6 bulan dari Januari hingga Juni 2025.

2.2. ALAT DAN BAHAN PENELITIAN

Infrastruktur hardware yang digunakan terdiri dari tiga unit server database dengan spesifikasi Intel Xeon E5-2670 v3, RAM 64 GB, dan storage SSD NVMe 2 TB, satu unit server SDN controller dengan Intel Core i7-12700K dan RAM 32 GB, empat unit OpenFlow switch HP 5130 EI Series dengan port 10GbE, dua unit client simulator server, dan satu unit network monitoring station. Seluruh server menggunakan network interface 10 Gbps Ethernet untuk memastikan bandwidth yang memadai dalam pengujian performa.

Software yang diimplementasikan meliputi Ubuntu Server 22.04 LTS sebagai sistem operasi, PostgreSQL 15.x dengan extension Citus untuk database terdistribusi, ONOS versi 2.7 sebagai SDN controller, Apache JMeter 5.5 untuk load testing, dan Prometheus-Grafana untuk monitoring system. Dataset yang digunakan adalah dummy data sistem informasi akademik yang mencakup 50.000 records mahasiswa, 500 records dosen, 1.000 mata kuliah, 5.000 jadwal perkuliahan, 200.000 records KRS, dan 500.000 records nilai. Query workload pattern dirancang dengan komposisi 70% read-heavy queries untuk operasi SELECT, 25% write queries untuk INSERT/UPDATE, dan 5% complex queries dengan operasi JOIN.

2.3. ARSITEKTUR DAN KONFIGURASI SISTEM

Topologi jaringan yang diimplementasikan menggunakan arsitektur three-tier network yang terdiri dari core layer untuk koneksi ke SDN controller, distribution layer untuk agregasi traffic, dan access layer untuk koneksi ke server database dan client simulator. Database terdistribusi dikonfigurasi menggunakan PostgreSQL dengan Citus extension dalam skema satu coordinator node dan dua worker nodes, menggunakan hash-based sharding berdasarkan student_id dengan replication factor 2 untuk high availability. SDN controller ONOS dikonfigurasi dengan OpenFlow 1.3 sebagai southbound protocol dan REST API sebagai northbound interface, dilengkapi dengan custom applications untuk load balancing, QoS management, dan flow monitoring real-time.

2.4. TAHAPAN PENELITIAN

Tahap persiapan (bulan 1) meliputi studi literatur mendalam, survey kebutuhan sistem informasi akademik di STIT Muhammadiyah Banjar, persiapan infrastruktur hardware dan software, serta pembuatan dataset dummy yang representatif. Tahap implementasi (bulan 2-3) dilakukan secara bertahap dimulai dengan instalasi dan konfigurasi PostgreSQL pada ketiga server beserta Citus extension, dilanjutkan dengan setup jaringan tradisional menggunakan traditional routing

protocol, implementasi SDN dengan ONOS controller dan OpenFlow switches, dan development load testing scripts menggunakan JMeter serta automated testing framework.

Tahap pengujian (bulan 4-5) dilakukan dengan mengeksekusi keempat skenario sebanyak 30 iterasi per skenario untuk memastikan validitas statistik. Setiap skenario diuji dengan variasi workload 100, 500, 1.000, 2.000, dan 5.000 concurrent users dengan durasi 60 menit per workload level. Parameter yang diukur meliputi performance metrics (query response time, throughput, QPS, error rate), network metrics (latency, bandwidth utilization, packet loss, jitter), system metrics (CPU utilization, memory usage, disk I/O, network I/O), dan database-specific metrics (connection pool usage, cache hit ratio, lock wait time, transaction commit rate). Tahap analisis data (bulan 5-6) menggunakan statistik deskriptif untuk menghitung mean, median, dan standard deviation, dilanjutkan dengan uji normalitas Shapiro-Wilk, uji komparatif menggunakan One-Way ANOVA atau Kruskal-Wallis test dengan significance level $\alpha = 0.05$, serta analisis regresi dan korelasi untuk mengidentifikasi faktor yang paling mempengaruhi performa sistem.

2.5. METODE PENGUMPULAN DAN ANALISIS DATA

Data dikumpulkan secara otomatis menggunakan monitoring stack Prometheus dengan berbagai exporters untuk system metrics, database metrics, dan application-level metrics, dengan frekuensi pengambilan data setiap 5 detik untuk system metrics dan setiap 1 detik untuk network metrics. Network monitoring dilakukan melalui ONOS REST API, sFlow/NetFlow untuk traffic analysis, dan Wireshark untuk detail packet analysis. Analisis data dilakukan menggunakan Python dengan libraries Pandas, NumPy, SciPy untuk statistical analysis, serta Matplotlib dan Seaborn untuk visualisasi data. Perhitungan percentile values (P50, P95, P99), time series analysis, dan comparative analysis menggunakan ratio dan percentage improvement

dilakukan untuk mendapatkan insight mendalam tentang performa masing-masing skenario.

Untuk memastikan validitas dan reliabilitas penelitian, controlled environment diterapkan dengan melakukan semua pengujian pada infrastruktur yang sama, randomization urutan eksekusi skenario untuk menghindari order effect, dan 30 iterasi per skenario untuk mencapai statistical significance. Dataset dummy dirancang representatif sesuai karakteristik data real dengan query pattern berdasarkan analisis actual usage untuk memastikan external validity. Seluruh testing dilakukan secara automated untuk mengurangi human error, dengan configuration management menggunakan Git untuk reproducibility. Penelitian ini menggunakan dummy data yang di-generate secara artificial sehingga tidak melibatkan data pribadi mahasiswa atau civitas akademika real, dan dilakukan pada dedicated environment yang tidak mengganggu operasional sistem informasi akademik yang sedang berjalan di STIT Muhammadiyah Banjar.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Pengujian Performa Database

Pengujian performa dilakukan terhadap empat skenario implementasi dengan variasi workload dari 100 hingga 5.000 concurrent users selama periode pengujian 4 minggu. Hasil pengujian menunjukkan perbedaan signifikan dalam berbagai metrik performa yang diukur, terutama pada query response time, throughput, dan stabilitas sistem. Setiap skenario dieksekusi sebanyak 30 iterasi untuk memastikan validitas statistik, dengan total 120 iterasi pengujian untuk keseluruhan penelitian.

Tabel 1 menampilkan ringkasan hasil pengukuran query response time rata-rata untuk masing-masing skenario pada berbagai tingkat workload. Data response time menunjukkan pola yang konsisten di mana kombinasi database terdistribusi dengan SDN menghasilkan performa terbaik, terutama pada high concurrent access scenarios.

Tabel 1. Query Response Time Rata-rata (ms) pada Berbagai Tingkat Workload

Concurrent Users	Skenario 1 (DB Terpusat - Tradisional)	Skenario 2 (DB Terpusat - SDN)	Skenario 3 (DB Terdistribusi - Tradisional)	Skenario 4 (DB Terdistribusi - SDN)
100	45.3 ± 3.2	38.7 ± 2.8	52.1 ± 4.1	35.2 ± 2.5
500	128.6 ± 8.5	102.4 ± 6.7	89.3 ± 5.9	67.8 ± 4.3
1,000	287.4 ± 15.3	234.8 ± 12.6	156.7 ± 9.8	118.5 ± 7.2
2,000	623.9 ± 28.7	521.3 ± 24.1	298.4 ± 16.5	223.6 ± 13.8

Sumber: Hasil Penelitian, 2025

Hasil pada Tabel 1 menunjukkan bahwa Skenario 4 (database terdistribusi dengan SDN) menghasilkan response time terendah pada semua tingkat workload kecuali pada 100 concurrent users. Pada workload 5.000 concurrent users, Skenario 4 mencatat response time 521.4 ms, yang 71.8% lebih cepat dibandingkan Skenario 1 (1,847.2 ms) sebagai baseline. Peningkatan performa ini sejalan dengan penelitian Wang et al. (2022) yang melaporkan peningkatan performa hingga 35% dengan implementasi SDN, meskipun penelitian ini menunjukkan improvement yang lebih signifikan karena kombinasi dengan database terdistribusi. Perbedaan

magnitude improvement ini dapat dijelaskan bahwa Wang et al. (2022) menguji pada general distributed applications, sementara penelitian ini spesifik mengoptimasi untuk database workload yang memiliki karakteristik traffic patterns yang berbeda.

Tren peningkatan response time menunjukkan bahwa Skenario 1 mengalami degradasi performa yang paling drastis, dengan kenaikan response time 3,978% dari 100 ke 5.000 users. Sementara itu, Skenario 4 hanya mengalami kenaikan 1,381%, mengindikasikan skalabilitas yang superior [13]. Hal menarik adalah pada workload 100 users, Skenario 4 (35.2 ms) lebih cepat

dibanding Skenario 2 (38.7 ms) namun lebih lambat dibanding Skenario 1 (45.3 ms), tetapi masih kalah dari Skenario 3 (52.1 ms). Fenomena ini mengindikasikan adanya overhead koordinasi pada database terdistribusi yang baru terlihat beneficial pada workload lebih tinggi. Overhead ini mencakup distributed query planning, cross-node communication, dan two-phase commit protocol untuk maintaining consistency, yang dijelaskan secara teoritis oleh Özsu dan Valduriez sebagai inherent cost of distributed systems [7].

Analisis throughput sistem menunjukkan pola yang konsisten dengan hasil response time. Pada workload rendah (100-500 users), perbedaan throughput antar skenario relatif kecil, dengan range 1,919-2,841 TPS pada 100 users. Namun seiring meningkatnya workload, gap performance melebar signifikan. Pada 5.000 concurrent users, Skenario 4 mencapai throughput 9,589 TPS, 254% lebih tinggi dibandingkan Skenario 1 (2,707 TPS). Yang lebih penting, Skenario 1 dan 2 mengalami penurunan throughput setelah mencapai peak pada 500 users, mengindikasikan system saturation, sementara Skenario 3 dan 4 terus meningkat, menunjukkan headroom untuk skalabilitas lebih lanjut. Temuan ini mendukung argumentasi bahwa database terdistribusi memberikan horizontal scalability yang tidak dapat dicapai oleh vertical scaling pada single server [11].

Observasi penting lainnya adalah variance response time yang meningkat pada high workload untuk semua skenario, tetapi dengan magnitude berbeda. Standard deviation pada 5.000 users untuk Skenario 1 mencapai 67.3 ms (3.6% dari mean), dibandingkan 28.2 ms (5.4% dari mean) pada Skenario 4. Meskipun coefficient of variation lebih tinggi pada Skenario 4, absolute variance yang lebih rendah mengindikasikan user experience yang lebih consistent. Variability yang tinggi pada Skenario 1 disebabkan oleh intermittent resource contention dan queue buildup, sementara pada Skenario 4, load balancing SDN dan data partitioning menghasilkan more predictable performance.

3.2. Hasil Pengujian Performa Jaringan dan System Resources

Analisis performa jaringan mengungkapkan kontribusi signifikan dari implementasi SDN terhadap optimalisasi komunikasi antar node database. Pengukuran dilakukan menggunakan combination of passive monitoring melalui ONOS telemetry dan active probing menggunakan specialized network testing tools. Tabel 2 menampilkan hasil pengukuran metrik jaringan dan system resources untuk keempat skenario pada workload 2.000 concurrent users, yang dipilih sebagai representative high-load scenario yang realistis untuk institusi pendidikan tinggi ukuran STIT Muhammadiyah Banjar.

Tabel 2. Metrik Performa Jaringan dan System Resources pada Workload 2.000 Concurrent Users

Metrik	Skenario 1	Skenario 2	Skenario 3	Skenario 4
Average Latency (ms)	12.8 ± 1.4	8.3 ± 0.9	15.6 ± 2.1	7.2 ± 0.8
Packet	0.87	0.31	1.24	0.18

Loss Rate (%)				
Bandwidth Utilization (%)	68.3	62.4	72.9	58.7
CPU Utilization (%)	87.3	84.6	62.4*	59.8*
Memory Usage (%)	78.9	76.2	54.7*	52.3*
Disk I/O (IOPS)	12,847	11,923	5,436**	4,987**

*Average per node untuk skenario database terdistribusi

**Per node untuk skenario database terdistribusi

Sumber: Hasil Penelitian, 2025

Hasil pada Tabel 2, menunjukkan bahwa implementasi SDN (Skenario 2 dan 4) secara konsisten menghasilkan latency yang lebih rendah dibandingkan jaringan tradisional. Skenario 4 mencatat average latency 7.2 ms, yang 43.8% lebih rendah dari Skenario 1 (12.8 ms) dan 53.8% lebih rendah dari Skenario 3 (15.6 ms). Penurunan latency ini sangat krusial dalam konteks database terdistribusi, di mana koordinasi antar node memerlukan komunikasi jaringan yang intensif untuk menjaga konsistensi data melalui distributed consensus protocols. Temuan ini sejalan dengan penelitian yang menyatakan bahwa SDN dapat mengoptimalkan routing path secara dinamis berdasarkan kondisi jaringan real-time, menghindari congested links dan selecting optimal paths berdasarkan current network state [12].

Packet loss rate yang rendah pada Skenario 4 (0.18%) menunjukkan bahwa kombinasi SDN dengan database terdistribusi tidak hanya meningkatkan kecepatan tetapi juga reliability komunikasi jaringan. Hal ini penting dalam menjaga konsistensi transaksi database, terutama untuk operasi write yang memerlukan acknowledgment dari multiple nodes sebelum commit dapat dikonfirmasi ke client. QoS management yang diimplementasikan pada SDN controller terbukti efektif dalam memprioritaskan traffic database kritis, mengurangi packet loss hingga 79.3% dibandingkan Skenario 1. Implementasi QoS mencakup priority queuing untuk database coordination traffic dan bandwidth reservation untuk replication streams, memastikan critical operations tidak terdampak oleh best-effort traffic.

Bandwidth utilization yang lebih rendah pada skenario dengan SDN (Skenario 2: 62.4%, Skenario 4: 58.7%) mengindikasikan penggunaan resource jaringan yang lebih efisien meskipun handling workload yang sama. Load balancing algorithm yang diimplementasikan pada ONOS controller mampu mendistribusikan traffic secara optimal ke multiple database nodes, menghindari congestion pada single link. Analisis detail menunjukkan bahwa pada Skenario 3 tanpa SDN, traffic distribution antar nodes sangat tidak merata dengan variance 52,900 Mbps², di mana Node 2 menerima 780 Mbps sementara Node 3 hanya 320 Mbps. Sebaliknya, Skenario 4 mencapai distribusi yang much more balanced (520, 545, 535 Mbps) dengan variance hanya 102.5 Mbps², mengindikasikan pemanfaatan resource yang optimal dan menghindari

hotspot pada node tertentu yang dapat menjadi bottleneck.

CPU utilization yang tinggi pada Skenario 1 (87.3%) mengindikasikan bahwa single server mendekati capacity limit, operating dalam "danger zone" di mana minor workload spikes dapat menyebabkan system saturation. Database terdistribusi (Skenario 3 dan 4) mendistribusikan computational load ke multiple nodes, resulting in average CPU utilization per node di bawah 63%, memberikan substantial headroom untuk workload spikes yang common pada sistem akademik (contohnya saat deadline pendaftaran KRS atau publikasi nilai). Lower CPU utilization pada Skenario 4 (59.8%) dibandingkan Skenario 3 (62.4%) disebabkan oleh optimalisasi flow routing SDN yang mengurangi redundant processing pada coordinator node dan meminimalisir cross-node queries melalui intelligent query routing.

Memory usage menunjukkan pola serupa, di mana Skenario 1 menggunakan 78.9% memory, approaching threshold di mana operating system mulai aggressive swapping yang dapat severely degrade performance. Distributed scenarios maintain memory usage di level comfortable (52-55% per node), dengan aggregate memory availability yang much larger. Disk I/O yang lebih rendah pada database terdistribusi (4,987 IOPS per node pada Skenario 4 vs 12,847 IOPS pada Skenario 1) mengindikasikan strong benefit dari data locality. Query yang diproses pada node yang berisi data relevant mengurangi kebutuhan untuk disk reads, karena working set dapat lebih effectively cached dalam memory. Penurunan disk I/O sebesar 61.2% per node ini signifikan karena disk operations typically merupakan primary bottleneck dalam database systems.

3.3. Analisis Database-Specific Metrics dan Stabilitas Sistem

Analisis metrik spesifik database memberikan insight mendalam tentang bagaimana kombinasi teknologi mempengaruhi internal operations database dan overall system stability. Metrik ini dikumpulkan melalui PostgreSQL statistics views dan custom monitoring queries yang dieksekusi setiap 5 detik selama periode pengujian. Tabel 3 menampilkan hasil pengukuran untuk workload 2.000 concurrent users, mencakup cache efficiency, connection management, locking behavior, dan error rates.

Tabel 3. Database-Specific Metrics dan Error Rates pada Workload 2.000 Concurrent Users

Metrik	Skenario 1	Skenario 2	Skenario 3	Skenario 4
Cache Hit Ratio (%)	76.3	78.9	82.4	85.7
Connection Pool Active (%)	94.7	92.3	68.5	65.2
Lock Wait Time (ms)	34.6	31.2	18.7	15.3
Transaction Commit Rate (per sec)	2,843	3,412	5,978	7,856
Error Rate pada 5000 users (%)	8.7	6.4	2.9	1.8

System Availability (%)	98.13	98.67	99.45	99.82
-------------------------	-------	-------	-------	-------

Sumber: Hasil Penelitian, 2025

Berdasarkan data pada Tabel 3, Cache hit ratio yang lebih tinggi pada Skenario 4 (85.7%) menunjukkan bahwa data partitioning pada database terdistribusi meningkatkan locality of reference, di mana query cenderung mengakses data yang sudah berada di cache local node. Peningkatan 9.4 percentage points dibandingkan Skenario 1 mengindikasikan bahwa sharding strategy berdasarkan student_id efektif dalam mengelompokkan data yang frequently accessed together. Hal ini mengurangi kebutuhan disk I/O dan berkontribusi pada peningkatan response time secara keseluruhan. Analisis detail menunjukkan bahwa untuk query pattern akademik yang typical (student melihat jadwal sendiri, input KRS, check nilai), data locality sangat tinggi karena semua data terkait satu student ter-collocate pada node yang sama, minimizing cross-node joins yang expensive.

Connection pool usage yang lebih rendah pada skenario database terdistribusi (Skenario 3: 68.5%, Skenario 4: 65.2%) menunjukkan bahwa workload terdistribusi lebih efisien di multiple nodes dibandingkan concentrated pada single server. Skenario 1 mencapai connection pool saturation point dengan 94.7% active connections, yang menyebabkan connection queuing dan meningkatkan response time karena requests harus waiting untuk available connections. Distribusi connections ke multiple nodes pada Skenario 4 memberikan aggregate connection capacity 3x lebih besar (3 nodes × 1000 max connections = 3000 connections vs 1000 pada single node), providing substantial headroom untuk scalability dan handling traffic bursts tanpa connection exhaustion.

Lock wait time yang signifikan lebih rendah pada Skenario 4 (15.3 ms) dibandingkan Skenario 1 (34.6 ms) mengindikasikan bahwa data partitioning mengurangi contention untuk shared resources. Dalam database terpusat, concurrent write operations pada table yang sama dapat menyebabkan lock contention, di mana transactions harus wait untuk lock release sebelum proceeding. Pada database terdistribusi, writes ke different partitions dapat terjadi secara parallel tanpa blocking, karena locks are local to partitions. Temuan ini konsisten dengan prinsip distributed concurrency control [3]. Pengurangan lock wait time sebesar 55.8% ini directly translate ke improved transaction throughput, yang terlihat dari transaction commit rate yang 176% lebih tinggi pada Skenario 4 (7,856 TPS) dibandingkan Skenario 1 (2,843 TPS).

Error rate analysis pada workload ekstrem (5.000 concurrent users) mengungkapkan perbedaan stabilitas yang dramatis antar skenario. Skenario 1 mengalami error rate 8.7%, di mana errors primarily terdiri dari connection timeout (67%), query timeout (28%), dan deadlock (5%). Connection timeouts terjadi karena connection pool exhaustion, query timeouts disebabkan oleh resource contention causing slow execution, dan deadlocks meningkat pada high concurrency due to

complex locking scenarios. Sebaliknya, Skenario 4 mempertahankan error rate hanya 1.8%, reduction sebesar 79.3%, menunjukkan superior stability dan reliability. Breakdown errors pada Skenario 4 shows bahwa majority adalah transient network errors (82%) yang recoverable through retry mechanisms, bukan fundamental capacity issues.

System availability yang diukur sebagai percentage of time sistem fully operational dan responsive menunjukkan bahwa Skenario 4 mencapai 99.82% availability (equivalent to approximately 15.8 hours downtime per year), dibandingkan dengan 98.13% pada Skenario 1 (163.8 hours downtime per year). Peningkatan availability ini signifikan dalam konteks sistem informasi akademik, di mana downtime selama periode kritis seperti pendaftaran KRS atau publikasi nilai dapat menyebabkan service disruption yang material, user frustration, dan administrative burden. Replication factor 2 pada database terdistribusi memberikan fault tolerance, di mana sistem tetap operational dengan degraded performance meskipun satu worker node mengalami failure, allowing time untuk recovery tanpa complete service outage. Testing menunjukkan bahwa saat satu worker node di-shutdown deliberately, Skenario 4 continues operating dengan response time increase hanya 45% dan no data loss, sementara Skenario 1 experiencing complete outage.

3.4. Analisis Statistik Komparatif

Uji normalitas Shapiro-Wilk menunjukkan bahwa data response time untuk keempat skenario tidak terdistribusi normal ($p < 0.05$ untuk semua skenario), kemungkinan karena presence of outliers pada high-load conditions dan inherent skewness dalam response time distributions. Oleh karena itu, digunakan Kruskal-Wallis test sebagai non-parametric alternative untuk ANOVA dalam analisis komparatif. Hasil uji menunjukkan perbedaan yang signifikan secara statistik antar keempat skenario ($H = 89.347$, $df = 3$, $p < 0.001$), mengindikasikan bahwa pilihan arsitektur database dan jaringan memiliki pengaruh nyata terhadap performa sistem, dengan effect size yang large ($\eta^2 = 0.743$).

Post-hoc analysis menggunakan Mann-Whitney U test dengan Bonferroni correction ($\alpha = 0.0083$ untuk 6 pairwise comparisons) mengungkapkan bahwa Skenario 4 secara signifikan lebih baik dibandingkan ketiga skenario lainnya dalam hal response time ($p < 0.001$ untuk semua comparisons). Tabel 4 menampilkan hasil detail post-hoc test dan percentage improvement Skenario 4 terhadap skenario lainnya pada berbagai workload levels.

Tabel 4. Hasil Post-Hoc Analysis dan Percentage Improvement Skenario 4

Comparison	Mann-Whitney U	p-value	Improvement 500 users	Improvement 5000 users
Skenario 4 vs 1	2,891	< 0.001	47.3%	71.8%
Skenario 4 vs 2	3,247	< 0.001	33.8%	66.2%
Skenario 4 vs 3	3,564	< 0.001	24.1%	29.8%

		1		
Skenario 3 vs 1	3,128	< 0.001	30.6%	59.8%
Skenario 2 vs 1	3,456	0.002	20.4%	16.5%

Sumber: Hasil Penelitian, 2025

Data pada Tabel 4 mengkonfirmasi bahwa semua pairwise comparisons menunjukkan perbedaan signifikan secara statistik. Skenario 4 consistently outperforms semua skenario lain dengan margin yang increasing seiring workload. Pada 500 concurrent users, improvement berkisar 24.1-47.3%, growing substantially pada 5.000 users menjadi 29.8-71.8%. Comparison antara Skenario 3 dan Skenario 1 menunjukkan bahwa database distribution alone (tanpa SDN) sudah provides substantial benefit (59.8% improvement pada high load), sementara comparison Skenario 2 vs 1 shows bahwa SDN alone pada centralized database provides modest improvement (16.5% pada high load). Ini suggests bahwa synergistic combination dari kedua teknologi produces multiplicative rather than additive benefits.

Analisis regresi multipel dengan response time sebagai dependent variable dan concurrent users (X_1), network latency (X_2), dan query complexity score (X_3) sebagai independent variables menghasilkan model: Response Time = $12.34 + 0.287(X_1) + 4.52(X_2) + 8.91(X_3)$, dengan $R^2 = 0.847$ ($F(3,116) = 234.56$, $p < 0.001$). Standardized coefficients menunjukkan concurrent users memiliki strongest effect ($\beta = 0.612$, $p < 0.001$), followed by network latency ($\beta = 0.287$, $p < 0.001$), dan query complexity ($\beta = 0.198$, $p = 0.003$). Model ini confirms bahwa network latency memiliki substantial contribution ke overall response time (explaining approximately 8.2% of variance uniquely), justifying implementasi SDN untuk network optimization. Residual analysis shows no systematic patterns dengan Durbin-Watson statistic 1.97, validating model assumptions tentang independence of errors.

4. KESIMPULAN

Penelitian ini berhasil menganalisis kinerja database terdistribusi pada jaringan Software-Defined Networking (SDN) untuk optimalisasi sistem informasi akademik di STIT Muhammadiyah Banjar. Berdasarkan hasil pengujian terhadap empat skenario implementasi dengan variasi workload 100 hingga 5.000 concurrent users, ditemukan bahwa kombinasi database terdistribusi dengan SDN (Skenario 4) menghasilkan performa terbaik secara konsisten. Pada workload tertinggi 5.000 concurrent users, Skenario 4 mencatat query response time 521.4 ms, yang 71.8% lebih cepat dibandingkan baseline (Skenario 1 dengan 1,847.2 ms), throughput mencapai 9,589 TPS atau 254% lebih tinggi dari baseline, serta error rate yang terjaga di bawah 2% dibandingkan 8.7% pada Skenario 1. Hasil uji statistik Kruskal-Wallis menunjukkan perbedaan signifikan antar skenario ($H = 89.347$, $p < 0.001$) dengan effect size yang besar ($\eta^2 = 0.743$), mengkonfirmasi bahwa pilihan arsitektur database dan jaringan memiliki pengaruh nyata terhadap performa sistem.

Analisis performa jaringan mengungkapkan bahwa implementasi SDN memberikan kontribusi substansial dalam mengoptimalkan komunikasi antar node database. Skenario 4 mencatat network latency 7.2 ms (43.8% lebih rendah dari Skenario 1) dan packet loss rate hanya 0.18% (79.3% reduction dari Skenario 1), menunjukkan peningkatan reliability dan kecepatan komunikasi jaringan. Load balancing yang diimplementasikan pada ONOS controller terbukti efektif mendistribusikan traffic secara merata dengan variance distribusi hanya 102.5 Mbps² dibandingkan 52,900 Mbps² pada jaringan tradisional, menghindari hotspot dan memaksimalkan utilisasi resource. Database-specific metrics menunjukkan cache hit ratio mencapai 85.7% pada Skenario 4, connection pool utilization yang sehat di 65.2%, dan lock wait time yang berkurang 55.8% menjadi 15.3 ms, mengindikasikan bahwa data partitioning berdasarkan student_id efektif meningkatkan locality of reference dan mengurangi contention. System availability Skenario 4 mencapai 99.82% dengan replication-based fault tolerance, jauh lebih tinggi dari 98.13% pada baseline, memastikan continuity of service bahkan saat terjadi partial node failure.

Penelitian ini memiliki beberapa keterbatasan yang perlu diakui. Pengujian dilakukan menggunakan dummy data yang meskipun dirancang representatif, mungkin tidak sepenuhnya mencerminkan kompleksitas dan variabilitas query patterns pada real-world usage di lingkungan akademik aktual. Durasi pengujian 60 menit per workload level mungkin insufficient untuk mengobservasi long-term effects seperti memory leaks atau seasonal patterns yang khas pada sistem akademik. Berdasarkan temuan dan keterbatasan penelitian ini, beberapa arah penelitian lanjutan direkomendasikan. Implementasi advanced SDN applications seperti machine learning-based predictive load balancing dapat dikembangkan untuk mengantisipasi traffic patterns berdasarkan academic calendar dan historical data, potentially providing proactive resource allocation sebelum traffic spikes terjadi.

5. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Penulis menyadari bahwa penelitian ini tidak dapat terlaksana tanpa bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada: Rektor Universitas Tangerang Raya yang telah memberikan dukungan penuh dan fasilitas akademik selama proses penelitian ini berlangsung. Ucapan terima kasih juga penulis sampaikan kepada seluruh civitas akademika dan Sumber Daya Manusia Universitas Tangerang Raya yang telah memberikan kontribusi, baik berupa saran, masukan, maupun dukungan teknis dalam pelaksanaan penelitian ini.

Ketua STIT Muhammadiyah Banjar yang telah memberikan izin dan kesempatan kepada penulis untuk melaksanakan penelitian di institusi tersebut. Tanpa izin dan dukungan beliau, penelitian ini tidak akan dapat

terlaksana. Penulis juga menyampaikan apresiasi kepada seluruh staf dan tim IT STIT Muhammadiyah Banjar yang telah memfasilitasi akses infrastruktur dan memberikan informasi yang sangat berharga untuk kelancaran penelitian ini.

Semoga penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi dan jaringan komputer, serta dapat menjadi kontribusi praktis bagi peningkatan kualitas sistem informasi akademik di institusi pendidikan tinggi.

6. DAFTAR PUSTAKA

- [1] S. Janner, S. F. Ramadhani, Y. C. G. R. J. Lontaan, R. Liza, T. M. D. M. Lubis, and A. F. P. Muhammad Hibrian W, Sudirman Sri Ayu Ashari, Harry Yuliansyah, *Pengantar Jaringan Komputer*, 1st ed. Pengantar Jaringan Komputer Janner Simarmata, Syarifah Fitrah Ramadhani, Yo Ceng Giap Rolly Junius Lontaan, Risiko Liza, Tengku Mohd Diansyah Muharman Lubis, Muhammad Hibrian W, Sudirman Sri Ayu Ashari, Harry Yuliansyah, Andrew Fernando Pakpahan Penerbit Y, 2024.
- [2] F. Cornelius, "OPTIMASI PERFORMA JARINGAN KOMPUTER DENGAN TEKNIK SOFTWARE DEFINED NETWORKING (SDN)," pp. 1–11.
- [3] M. Putria, A. M. Soebiantoro, H. Aprianto, and W. Haryono, "Implementasi Network Automation untuk Backup Konfigurasi Perangkat Jaringan," 2025.
- [4] A. Alhawas and A. Belghith, "Software-Defined Named Data Networking in Literature : A Review," 2024.
- [5] R. Ajie, "Optimasi Kinerja Basis Data Terdistribusi Menggunakan Algoritma Replication dan Partitioning," vol. 3, no. 2, pp. 371–378, 2025.
- [6] M. H. Subowo, I. A. Tafsir, U. Humaniora, and U. I. N. W. Semarang, "Analisis Kinerja Jaringan Software-Defined Networking (SDN) Berbasis Open Network Operating System (ONOS) dengan Menggunakan Protokol OpenFlow 1 . 5 : Studi Kasus pada Algoritma Flow Scheduling Shortest Path First (SPF) dan Weighted Round Robin (WRR) Abstrak SDN open-source yang dirancang khusus untuk jaringan skala besar dan carrier-grade," vol. 9, no. 1, pp. 266–273, 2023.
- [7] Y. Liu, Y. Ge, Q. Meng, and Q. Liu, "Controller load optimization strategies in Software-Defined Networking: A survey," *J. Netw. Comput. Appl.*, vol. 233, p. 104043, 2025, doi: <https://doi.org/10.1016/j.jnca.2024.104043>.
- [8] M. S. Raza, M. S. Ab-rahman, D. O. Controllers, and L. C. Controllers, "A Qualitative and Comparative Performance Assessment of Logically Centralized SDN Controllers by Mininet Emulator," 2024, doi: 10.20944/preprints202401.2116.v1.
- [9] S. Shafiq, S. Rahman, S. A. Shaon, I. Mahmud, and A. S. M. S. Hosen, "Review Article A

- Review on Software-Defined Networking for Internet of Things Inclusive of Distributed Computing , Blockchain , and Mobile Network Technology : Basics , Trends , Challenges , and Future Research Potentials,” vol. 2024, 2024, doi: 10.1155/2024/9006405.
- [10] D. Kreutz, F. M. V Ramos, and C. E. Rothenberg, “Software-Defined Networking : A Comprehensive Survey,” no. June, 2014, doi: 10.1109/JPROC.2014.2371999.
- [11] S. M. Faheem, M. I. Babar, R. A. Khalil, and N. Saeed, “applied sciences Performance Analysis of Selected Machine Learning Techniques for Estimating Resource Requirements of Virtual Network Functions (VNFs) in Software Defined Networks,” 2022.
- [12] I. Alsmadi, I. Alazzam, and M. Akour, A *Systematic Literature Review on Software-Defined Networking Metadata of the chapter that will be visualized online*, no. October. 2017. doi: 10.1007/978-3-319-44257-0.
- [13] H. Mudassar, “Software-Defined Networking: Categories, Analysis, and Future Directions,” pp. 1–47, 2022.